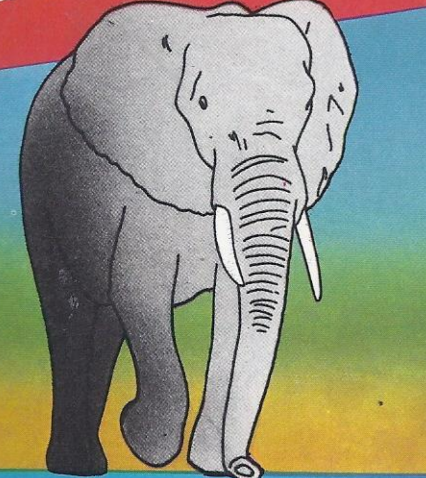
commodore Magazine

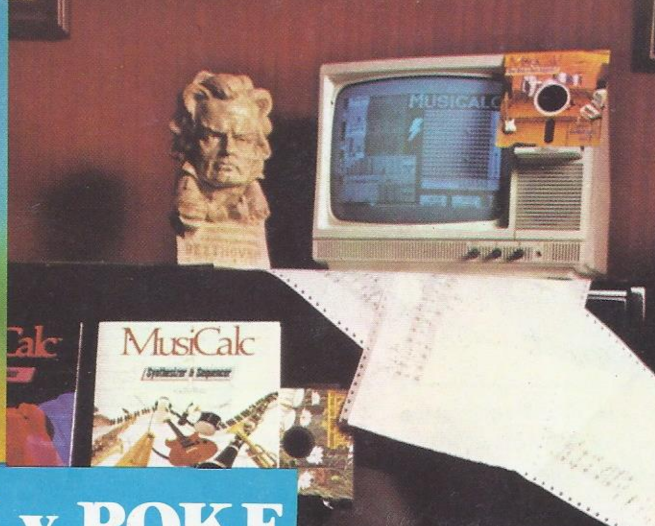
AÑO I - Núm. 7 - Septiembre 1984 - 250 Ptas.

REVISTA INDEPENDIENTE PARA USUARIOS

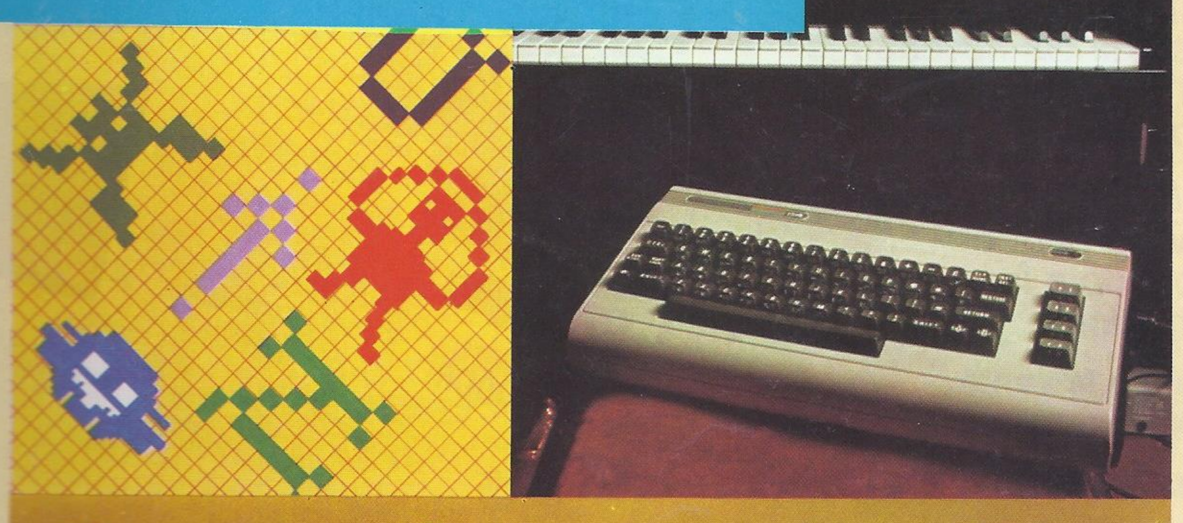
Ampliacion de memoria para el Vic-20

Un virtuoso ordenador





Graficos con PEEK y POKE

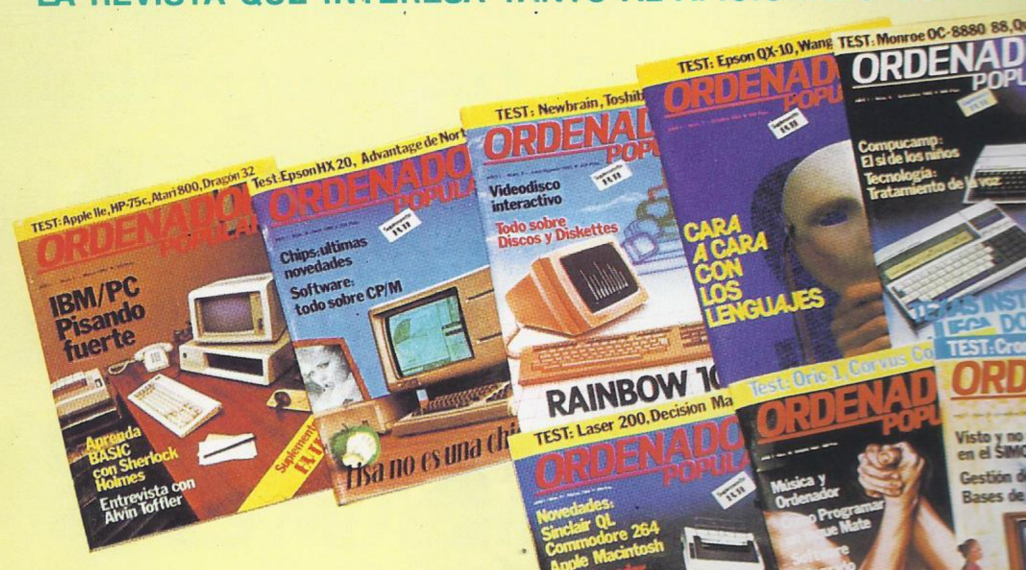


Concurso, juegos y aplicaciones

La versión española de Popular Computing

ORDENADOR POPULAR

LA REVISTA QUE INTERESA TANTO AL AFICIONADO COMO AL PROFESIONAL





Una publicación que informa con amenidad acerca de las novedades en el campo de las computadoras personales.

ORDENADOR POPULAR, la revista para el aficionado a la informática.

Ya está a la venta

Cómprela en su kiosco habitual o solicítela a:

ORDENADOR POPULAR

EDISA,
López de Hoyos, 141,
28002 Madrid

commodore Magazine

Sumario

Commodore Magazine es una publicación de Publinformática, S. A., C/ Bravo Murillo, 377 - 28020 Madrid Tel. (91) 733 74 13 / 47 / 63 / 97.
REDACCION
Director:
Alejandro Diges.
Colaboradores:
Aníbal Pardo.
Gumersindo García.
Roberto Menéndez.
Simeón Cruz.
Fernando García.
Manuel Arias.
Diseño:
Ricardo Segura.
EDITORIAL
Presidente:
Fernando Bolín.
Director Editorial:
Norberto Gallego.
Coordinador Editorial:
J. A. Sanz.
ADMINISTRACION
Gerente de Circulación y Ventas:
Luis Carrero.
Suscripciones: Antonio Zurdo.
Producción:
Miguel Onieva.
Publicidad Madrid:
Roberto Rodríguez.
Bravo Murillo, 377.
Madrid-20.
Tel. (91) 733 74 13.
Publicidad Barcelona:
Pelayo, 12.
Tel. (93) 301 47 00, Ext. 27.
Distribuye: SGEL.
Avda. Valdelaparra s/n.
Alcobendas, Madrid.
Imprime: Novograph, S. A.,
Ctra. de Irún, Km. 12.450
Madrid.
Fotomecánica: Karmat. Pantoja, 10. Madrid.
Depósito Legal: M-6622-1984.

Año 1
Num. 7

SUSCRIPCIONES

Rogamos dirija toda la correspondencia relacionada con suscripciones o números atrasados a:

commodore Magazine
EDISA
López de Hoyos, 141, 5.º
MADRID-2
Tel. 415 97 12

Esta revista no mantiene relación de dependencia de ningún tipo con respecto de los fabricantes de ordenadores Commodore Business Machines ni de sus representantes.

Editorial

Ya estamos casi todos nuevamente incorporados al trabajo. El bronce de la piel tiene un último destino: el teclado del ordenador.

Esta edición de **Commodore Magazine** es algo densa en número de artículos, pero tampoco se han olvidado las tradicionales secciones de Programas y Concurso.

Por necesidades de espacio lamentamos haber tenido que posponer la segunda parte de la serie "El Misterio del BASIC", que obligatoriamente ha tenido que esperar a octubre; igual que "Como diseñar juegos para ordenador".

Una modificación, la tabla de claves para teclear correctamente los programas ha sido ampliada notablemente, incluyéndose símbolos de utilización no muy corriente, pero que es necesario conocer.

El MusiCalc es una "hoja de trabajo" para escribir, editar y reproducir música con el **Commodore 64.** A este paquete comercial le hemos dedicado una parte de la revista, describiendo las más importantes de sus múltiples características. Es en realidad una flexible herramienta de enormes recursos. Podemos decir que hemos disfrutado comprobando su funcionamiento.

Continuando con la serie que trata de la otra forma de leer el Manual, para sacar mayor provecho al **C-64,** entramos con los gráficos programables desde el BASIC mediante PEEK y POKE. Le acompañan dos interesantes programas explicativos.

La segunda parte del artículo sobre el diseño y utilización de los lápices ópticos también tiene cabida en esta ocasión.

El montaje está destinado al **Vic-20.** Se trata de una ampliación de memoria de 3 Kbytes, de gran utilidad para poder desarrollar programas de mayor longitud.

A quienes guste teclear encontrarán suficientes programas para llenar sus horas libres.

Por último, un colaborador de **Commodore Magazine** nos ha remitido un Programa Monitor para el **64,** desarrollado en BASIC, que puede ser de utilidad a quienes quieran iniciarse en el lenguaje máquina y no quieran gastar demasiado en la compra de un paquete adecuado. De cualquier manera, el segundo será más rápido y tendrá otras utilidades.



Códigos de control para el VIC-20 y el C-64

COMO SE VE	COMO SE TECLEA	EFECTO CONSEGUIDO
COLORES DEL VIC-20 Y DEL 64		
	CTRL+1	NEGRO
	CTRL+2	BLANCO
	CTRL+3	ROJO
	CTRL+4	CIAN
	CTRL+5	PURPURA
	CTRL+6	VERDE
	CTRL+7	AZUL
	CTRL+8	AMARILLO

COMO SE VE	COMO SE TECLEA	EFECTO CONSEGUIDO
COLORES DEL 64 SOLAMENTE		
	CBM+1	NARANJA
	CBM+2	MARRON
	CBM+3	ROSA
	CBM+4	GRIS OSCURO
	CBM+5	GRIS MEDIO
	CBM+6	VERDE CLARO
	CBM+7	AZUL CLARO
	CBM+8	GRIS CLARO

COMO SE VE	COMO SE TECLEA	EFECTO CONSEGUIDO
CODIGOS DE CURSOR Y CONTROL		
	HOME	CURSOR A CASA
	SHIFT+HOME	LIMPIA PANTALLA
	CRSR	CURSOR IZQUIERDA
	SHIFT+CRSR	CURSOR DERECHA
	CRSR	CURSOR ABAJO
	SHIFT+CRSR	CURSOR ARRIBA
	CTRL+9	CARACTER INVERSO
	CTRL+0	CARACTER NORMAL
	DEL	BORRAR
	SHIFT+DEL	INSERTAR

COMO SE VE	COMO SE TECLEA
TECLAS DE FUNCION	
	F1
	F2=SHIFT+F1
	F3
	F4=SHIFT+F3
	F5
	F6=SHIFT+F5
	F7
	F8=SHIFT+F7

Por su diseño, el **Commodore 64** es uno de los microordenadores actualmente existentes que dispone de mayores posibilidades en la generación de sonidos. De hecho el ordenador se convierte en una herramienta para el aprendizaje y ejecución de música. Sin embargo, no es todo tan sencillo, en el repertorio de comandos utilizados por este modelo no aparece uno solo capaz de generar un tono sin mayores dificultades. Ello obedece a que el *chip* sintetizador, controlado desde el microprocesador, asimila la información que depositemos en determinadas posiciones de memoria. A este tipo de funcionamiento se le llama por mapa de memoria, pues accediendo a cada localización se controla una determinada posibilidad del dispositivo situado tras ella.

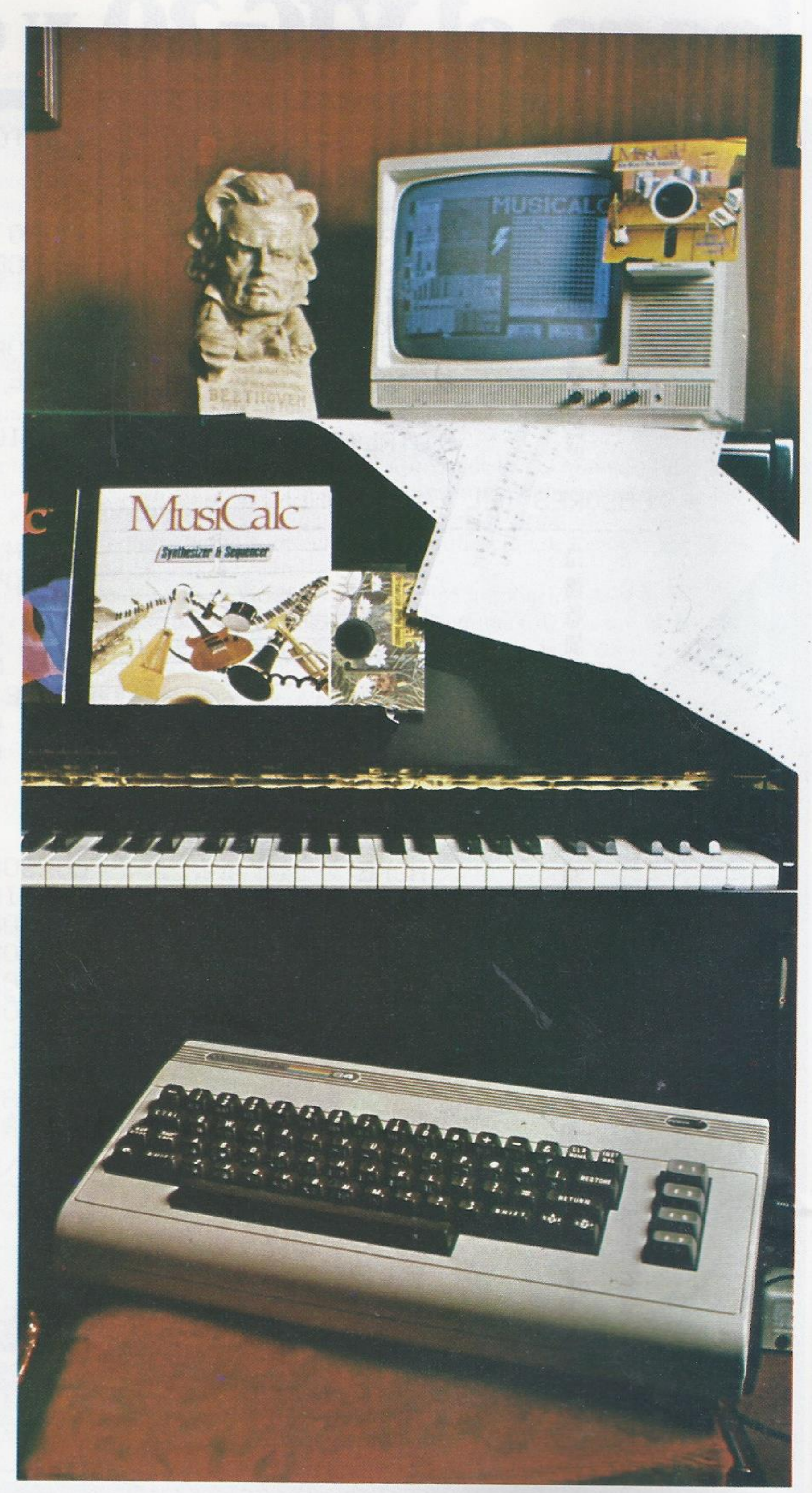
El *chip* es fabricado especialmente para **Commodore** por su firma subsidiaria **MOS Technology**, que lo bautizó como **6581 SID** *(Sound Interface Device)*. Se trata de un auténtico sintetizador musical, controlado totalmente por procedimientos digitales. En su interior dispone de todas las posibilidades existentes en una gran mayoría de sintetizadores musicales disponibles en el comercio.

La generación de los tonos iniciales se produce en 3 osciladores de tono, independientes entre sí, capaces de producir sonidos en la gama de 0 a 4 KHz en cuatro formas de onda distintas: triangular, diente de sierra, impulsos de anchura variable y ruido blanco. Cada oscilador corresponde a una de las tres voces que ofrece el sintetizador.

Los 3 osciladores pueden sincronizarse entre sí.

Una vez producida la señal primaria, ésta pasa directamente a 3 moduladores de amplitud, uno para cada voz. Un modulador de amplitud no es otra cosa que un amplificador de señal al que se le puede controlar la ganancia mediante una segunda señal. Esto no quiere decir otra cosa que modular el volumen instantáneo del tono a voluntad, bajo el control de una tensión eléctrica, que en nuestro caso es producida por los 3 generadores de envolvente.

La identidad de un sonido es defi-



MusiCalc:

EL ORDENADOR VIRTUOSO

nida en gran medida por la envolvente que modula al tono. Veamos por ejemplo que cuando simulamos una explosión con la boca lo que estamos haciendo es producir ruido blanco y luego modulamos su intensidad con el movimiento de los labios, haciendo que su volumen suba desde un mínimo a un máximo y después no lo anulamos, sino que disminuimos el volumen hasta su total extinción. Algo similar podríamos imaginar con un piano, la cuerda resulta golpeada de pronto, el volumen del sonido sube al máximo y después se mantiene hasta ahogarse.

Los generadores de envolvente del **6581** proporcionan una respuesta exponencial a los datos digitales aplicados. Recordemos que la envolvente de un sintetizador (en los sonidos naturales la envolvente puede ser algo más compleja) se divide en cuatro partes características. El período de **Ataque,** que puede establecerse entre 2 milisegundos y 8 segundos, es el tiempo empleado para aumentar la amplitud de la señal de audio desde cero al máximo. A continuación el **Decaimiento,** pues la señal no puede mantenerse físicamente mucho tiempo en el máximo, cayendo hasta una amplitud menor que puede durar más tiempo, a la que se llama **Sostenimiento.** El período de duración del **Decaimiento** puede programarse entre 6 milisegundos y 24 segundos, que se combina con el **Sostenimiento,** del que podemos controlar cual será su amplitud. Por último sólo queda la extinción de la señal, que puede ser fulminante o durar hasta otros 24 segundos. A este último se le llama período de **Relajación.**

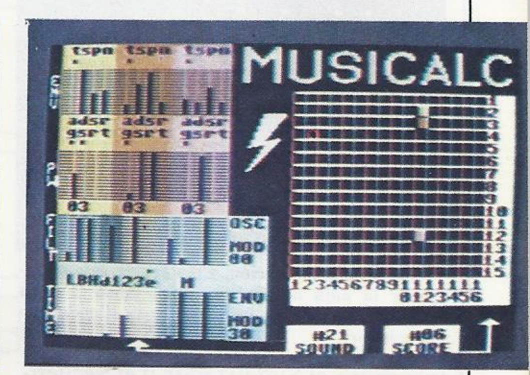
Posteriormente, las señales pueden aplicarse bien a un filtro programable o directamente al amplificador de salida.

Otra de las características que personalizan a los sonidos es su contenido en armónicos, es decir el número de otras frecuencias distintas a la fundamental que acompañan a ésta. La onda sinusoidal (senoidal) no se encuentra en la naturaleza, tiene que ser siempre generada electrónicamente. Según el teorema que en su día desarrolló **Fourier,** cualquier forma de onda puede ser descompuesta en una serie de ondas sinusoidales de diferente amplitud, donde la de menor frecuencia es la fundamental.

Aspecto que ofrecen el Manual y el diskette del Módulo 1.

Las 4 diferentes formas de onda generables en los 3 osciladores del **6581** no son una excepción. Se trata pues de diferentes combinaciones de ondas senoidales. Filtrándolas se puede cambiar la amplitud de sus frecuencias, variando la presencia de los diferentes armónicos.

Un filtro es un dispositivo que favorece el paso de una gama determinada de frecuencias y dificulta (o anula) otras. Los filtros se dividen en 4 grupos principales: **Pasa Altos,** que sólo dejan pasar las frecuencias superiores a una determinada (frecuencia de corte); **Pasa Bajos,** sólo deja pasar las frecuencias inferiores a la de corte; **Paso Banda,** facilitan el paso de las frecuencias situadas en una franja limitada por las frecuencias de corte superior e inferior y el de **Rendija,** que deja pasar todo excepto una franja de frecuencias. El **SID** dispone de los 3 primeros, utilizándose uno u otro por programación.



Panel correspondiente al Módulo 1: Sintetizador y Secuenciador.

El amplificador de salida es similar a los de modulación de amplitud, sólo que en este caso se establece un solo volumen de la señal compuesta de salida, no requiriendo generador de envolvente.

El **SID** lleva además incorporados 2 conversores A/D (analógico/digital) para conectarles sendos potenciómetros, que actúan como *joysticks* de juegos.

Existe otra característica casi oculta, de la que el fabricante apenas habla. Es posible atacar el **SID** con una señal externa, por ejemplo una guitarra eléctrica, pero el procedi-



MusiCalc

miento es delicado y podría traer graves deterioros en el ordenador si no se hace adecuadamente. En un artículo futuro trataremos esta posibilidad en **Commodore Magazine.** La señal externa puede ser filtrada y mezclada en el amplificador de salida.

La firma **Waveform** ha desarrollado un conjunto de programas que facilitan la utilización de las posibilidades musicales del **C-64.** El paquete ha sido comercializado bajo el nombre de **MusiCalc** y es una serie de varios *diskettes.* Con él se convierte al ordenador en la ansiada herramienta para aprendizaje de música, pero tiene avanzadas posibilidades que lo convierten en un útil de composición musical.

Los 3 *diskettes* principales son el **MusiCalc 1,** módulo principal que incorpora el Sintetizador y el Secuenciador. El Módulo 2 es llamado **ScoreWriter** (algo así como escritor de piezas) y el Módulo 3, *Keyboard Maker* (conformador del teclado), completan el trío.

Pantalla de composición. En ella se puede seleccionar la nota octava que corresponde a cada uno de los dieciséis pasos de cada fila del secuenciador.

Para animarnos y observar las enormes posibilidades que se abren ante nosotros, al conjunto le acompañan dos *diskettes* que contienen composiciones, uno de rock y el otro de ritmos africanos y latinos, pero en el Módulo 1 existen melodías más tradicionales. El Módulo 1 es el principal, con el que se facilita la utilización del **64** como Sintetizador. Sin embargo, hacer música no es sólo producir sonidos y por esto se le ha dotado del secuenciador, que mantiene los tiempos y advierte al ordenador cuando es el momento de producir la siguiente nota de una melodía. Este secuenciador va leyendo paso a paso cada una de las notas previamente introducidas para cada una de las tres voces.

La carga del Módulo 1 en el ordenador es tan lenta como es tradicional. Una vez introducido el *diskette* en la unidad correspondiente, necesita unos dos minutos y medio para cargarlo. Lo primero que aparece en la pantalla es el Panel, en el que aparece reflejado el estado de los distintos controles que afectan al sintetizador **SID.**

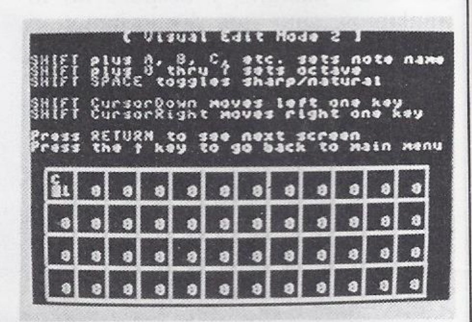
En la parte izquierda vemos dos tipos diferentes de controles: los conmutadores y los deslizantes. Los conmutadores aparecen representados como un pequeño cuadrado, si es visible está activado. Los otros, deslizantes, imitan a los potenciómetros utilizados en amplificadores y mesas de mezcla de sonido, o más bien a un termómetro, que suben o bajan la altura de su indicador. Cada una de las tres voces está representada en él. En la parte superior están los osciladores, donde cada forma de onda ha sido asociada a un conmutador. Inmediatamente debajo aparecen los doce controles continuos del generador de envolvente, cuatro para cada voz. Seguidamente los controles destinados a establecer la anchura de los impulsos producidos por cada oscilador, cuando es seleccionada esta opción.



Igualmente están representados los controles del filtro, las temporizaciones y los moduladores especiales.

En el lado derecho aparece visualizada una rejilla en la que están representadas las voces en forma de un cuadrado que ocupa un puesto en la rejilla. Estos cuadrados se mueven a medida que el secuenciador avanza un paso en la composición reproducida. Paralelamente al Panel existe un menú de opciones, al que se puede acceder en cualquier momento. En él están observados los diferentes accesos a la unidad de *diskettes,* bien para visualizar el directorio o para guardar y cargar composiciones. También permite acceder a una tercera pantalla, que es la de composición. Los cambios realizados en ella nos permiten componer y/o editar música.

Cuando aparece el Panel en la pantalla existen tres modos distintos de operación: Sintetizador, Composiciones pregrabadas y Teclado.



Aspecto que ofrece una pantalla del Módulo Keyboard Maker. Cada cuadrado representa una tecla, a la que se puede programar independientemente.

En el primero podemos ajustar el sonido variando directamente los valores de los controles antes descritos. Con las teclas de función F1 a F7 podemos establecer la posición de los controles deslizantes o la activación y desactivación de los conmutadores. Las teclas de cursor indican al ordenador cuál de las dos cosas queremos hacer. La forma de acceder a uno u otro control es muy sencilla, a cada fila de teclas se le hace corresponder una fila de controles de la pantalla. Así, presionando la tecla homóloga, podemos acceder a ese control mediante las teclas de cursor y de función. El siguiente modo, Composiciones pregrabadas, permite repro-

ducir melodías almacenadas en la memoria. Presionando la tecla SHIFT y una de un grupo de 32 del cuerpo principal del teclado obtenemos una melodía. Combinando una de esas teclas con la tecla **Commodore** conseguimos una diferente configuración del sintetizador, alterándose los controles del lado izquierdo de la pantalla.

El modo Teclado convierte al **C-64** en un piano, cuando se presiona una tecla suena la nota correspondiente.

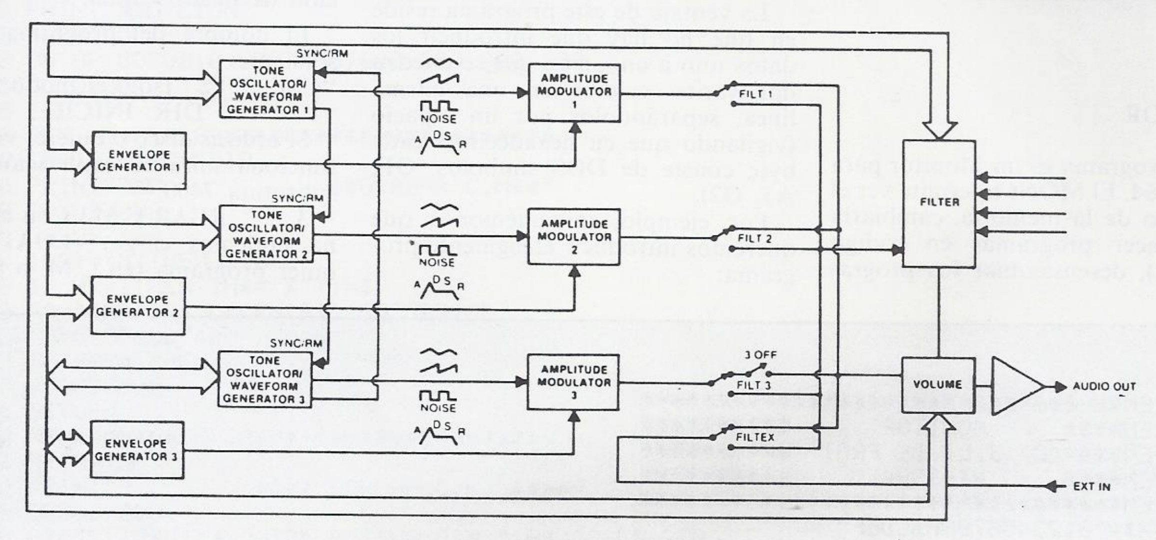
Volvamos sobre la pantalla de composición/edición de música. En el lado derecho aparecen las doce notas correspondientes a una octava cromática, utilizándose la notación sajona para la nota: C para do, D para re... B para si. Debajo hay otra columna de números entre el cero y el siete, que corresponden a la octava. Asociadas a ellas hay una fila de 16 rombos con la primera y 16 asteriscos con la segunda. Representan los 16 pasos que podemos programar mediante las teclas de cursor, que nos sitúan en la parte de arriba, para elegir la nota, o en la de abajo, para seleccionar en qué octava queremos situar la nota. La tecla de movimiento del cursor en sentido horizontal nos lleva a la nota siguiente o precedente, que aparece indicada por una cruz debajo, donde hacemos la misma operación, hasta completar las 16. Las teclas de función son nuevamente quienes sirven para ajustar nota y octava. Es de gran ayuda que la melodía que vamos componiendo se

Es igualmente posible controlar la duración relativa de las notas. Por ejemplo, un paso sería un dieciséisavo de nota, dos pasos un octavo, cuatro pasos un cuarto, etc.

El Módulo 2 contiene tres ampliaciones destinadas al Módulo 1. Trabajan solamente con ficheros de música pregrabada. La primera permite obtener las partituras en una impresora gráfica matricial.

La segunda posibilidad permite crear composiciones más largas y complicadas de lo que es posible sólo con el Módulo 1, es un "secuenciador de secuencias".

La tercera mejora sirve para controlar la señal paso a paso que controla al secuenciador con otro **C-64**, o incluso con un dispositivo



repita una y otra vez mientras componemos y editamos.

Como es lógico, podemos acceder a cada voz por separado a la hora de componer.

Pero pensamos que una vez compuesta una melodía de sólo 16 notas nos quedamos cortos. La respuesta es que podemos componer otro bloque y unirlo al anterior. De vuelta al Panel vemos que la rejilla dispone de 15 filas. Pues bien, a cada fila le hacemos corresponder un bloque de 16 notas. Así podemos utilizar para hacer que una composición se repita sucesivamente o encadenar varias paí sucesivamente o encadenar varias para una melodía larga, sin repeticiones.

externo.

Finalmente, el Módulo 3 es el Conformador del teclado. Permite configurar el teclado de la forma más cómoda para nuestras necesidades de ejecución. Es posible elegir entre 80 escalas diferentes previamente programadas en este *diskette*, trabajando con el Módulo 1.

En un principio no es fácil memorizar todas las posibilidades que ofrece el **MusiCalc**, pero el manual (en inglés por ahora) siempre tiene la respuesta, mejor el ejemplo, adecuado para salir del paso. Cuando se alcanza un cierto dominio en el manejo, el ordenador pasa a convertirse en algo de extrema utilidad musical.



Existe la posibilidad de afinar el ordenador para que actúe en compañía de otro instrumento.

Programa monitor para

Siempre resulta útil disponer de programas de utilidad que nos ayuden a perfeccionar nuestra programación. Cuando queremos emplear el lenguaje máquina la cosa se hace más tediosa. Con este programa ganamos diversas posibilidades. No obstante, su ejecución resulta un poco lenta, por haber sido desarrollado en BASIC, pero su utilidad es totalmente vigente y, sobre todo, resulta mucho más didáctico el análisis de su mecanismo de funcionamiento.

MONITOR

Este programa es un Monitor para el **CBM-64**. El **MON64** permite ver el contenido de la memoria, cambiarla (para hacer programas en código máquina), desensamblar los programas en C.M., grabarlos en cinta o *diskette*, ejecutarlos y cargar nuevos programas de una cinta o *diskette*.

En las líneas 100-200 está en menú de opciones que son:

1. ENSAMBLADOR: Esta opción permite variar el contenido de la memoria utilizando valores hexadecimales o decimales. Para escribir programas en código máquina no tenemos más que indicar la dirección de inicio y a continuación los códigos a introducir.

La ventaja de este programa reside en que no hay que introducir los datos uno a uno, sino que se pueden dar varios valores en una misma línea, separándolos por un espacio (vigilando que en hexadecimal cada byte conste de DOS símbolos: OF, A3, O2).

Por ejemplo, imaginémonos que queremos introducir el siguiente programa:

```
C000    LDA  #$04
C002    JSR  #FFC0
```

La pantalla quedaría como sigue (ver fig. 1). Después de pedirnos la dirección de inicio vamos introduciendo los códigos (ponemos M para volver al menú).

2. GRABAR C. M.: Esta opción graba (SAVE) en cinta el programa realizado con ENSAMBLADOR o también cualquier zona de la memoria. Sólo hemos de indicar la dirección de inicio y final.

El nombre del programa será el siguiente:

"CM" + "S" (sólo el modo Hex.) + DIR. INICIO

Si utilizas disco o quieres ver cómo funciona mira la explicación de la subrutina 7400.

3. CARGAR C.M.: Con esta rutina podemos cargar (LOAD) cualquier programa (en C.M. o borraría

```
0 REM*********************************
1 REM****     MONITOR      **********
2 REM****(C) J.L. DE PRAT  **********
3 REM****     VI-1984      **********
4 REM*********************************
5 HX$="0123456789ABCDEF"
6 DIMV$(80),V(80),M$(152,2),M(152)
7 PRINT"]]]] ESPERE."
8 GOSUB8500
9 GOTO 100
10 PRINT"]"
20 PRINT"          *** ENSAMBLADOR ***"
25 GOSUB7000
40 PRINT"]]DIR. INICIO?"
50 INPUTDIR$:IFDIR$=""THEN GOTO50
51 IFHX=1THENH$=DIR$:ER=0:GOSUB5000:DIR=D:GOTO53
52 DIR=VAL(DIR$)
53 PRINT:IDIR=DIR
55 IFER=1THENGOTO40
60 V$="":PRINTDIR$;:INPUTV$:IFV$=""THENGOTO60
64 IF V$="FIN"ORV$="M"THEN FDIR=DIR:RETURN
65 GOSUB4500
70 IF ER=1THEN PRINT"ERROR"
```

```
80 GOTO60
100 REM*****PAGINA   MENU***********
105 PRINT"♥ "
110 PRINT"          ***  MENU INICIAL ***"
115 PRINT"▼▼▶▶▶▶▶▶1. ENSAMBLADOR"
120 PRINT"▶▶▶▶▶▶2. GRABAR C.M."
130 PRINT"▶▶▶▶▶▶3. CARGAR C.M."
140 PRINT"▶▶▶▶▶▶4. VOLCADO"
150 PRINT"▶▶▶▶▶▶5. EJECUTAR C.M."
155 PRINT"▶▶▶▶▶▶6. MNEMONICOS"
160 PRINT"▶▶▶▶▶▶7. FIN"
170 PRINT"▼▼OPCION?"
180 INPUT OP
190 ON OP GOSUB10,300,450,600,800,900,200
195 GOTO100
200 END
300 REM ***GRABAR**********************
310 CM=IDIR:AC=FDIR
320 PRINT"♥▼▼          **GRABAR C.M**"
330 GOSUB7000
340 GOSUB7100
362 IFA$="M"THEN 390
379 IF HX=1THEN IN$="$"+IN$
380 A$="CM"+IN$:L=LEN(A$):GOSUB8200
382 LO=0:GOSUB7400
385 IF ERR=1THEN300
390 GOSUB7500
400 RETURN
450 REM******CARGA*******************
460 PRINT"♥ "
470 PRINT"          **** CARGAR C.M. ****"
475 GOSUB7000
480 A$="":INPUT"▼NOMBRE RUTINA C.M.";A$:GOSUB8200
490 INPUT"▼▼▼DIRECCION DE CARGA";D$:IFHX=0THENGOTO497
495 H$=D$:GOSUB5000:DC=D:GOTO500
497 DC=VAL(D$)
500 LO=1:GOSUB7400
510 IF ERR=1THEN GOTO450
520 GOSUB7500
530 RETURN
600 REM***VOLCADO*********************
605 CM=IDIR:AC=FDIR
610 PRINT"♥"
615 PRINT"         ****VOLCADO DE MEMORIA****"
620 GOSUB7000
630 GOSUB7100
640 GOSUB7300
680 IFA$="M"THEN 770
700 PRINT"▼PULSA 'RETURN' PARA ACABAR"
701 PRINT"O LA BARRA PARA PARAR.▼▼▼"
```

Programa monitor para

```
705 FORME=CM TO AC
710 IFHX=1THEND=ME:GOSUB5500:P$=H$:GOTO720
715 P$=STR$(ME)
720 PRINTP$;"    ";
730 C=PEEK(ME):IFHX=1THEND=C:GOSUB5500:C$=H$:GOTO740
735 C$=STR$(C)
740 PRINTC$
745 GETA$:IFA$=CHR$(13)THENME=AC
746 IFPEEK(197)=60THENGOTO746
750 NEXT
760 PRINT#1:CLOSE1:GOSUB7500
770 RETURN
800 REM **EJECUCION********************
805 CM=IDIR
810 PRINT"□■            **** EJECUCION ****"
815 GOSUB7000:D=CM:GOSUB5500:IN$=H$
820 PRINT"■DIR. INICIO: ";CM,"$";IN$
830 INPUT"DIRECCION DE LLAMADA";LL$:IF HX=1THENH$=LL$:GOSUB5000:LL=D:GOTO840
835 LL=VAL(LL$):D=LL:GOSUB5500:LL$=H$
840 PRINT"■LLAMADA (SYS): "LL,"$";LL$
850 PRINT"■PULSA 'RETURN' PARA EJECUTAR EL C.M"
860 GETA$:IFA$<>CHR$(13)ANDA$<>"M"THEN860
861 IFA$="M"THEN890
870 SYS LL
875 GOSUB7500
890 RETURN
900 REM****MNEMONICOS*****************
910 PRINT"□"
915 PRINT"            ***MNEMONICOS***"
918 CM=IDIR:AC=FDIR
920 GOSUB7000
930 GOSUB7100
940 GOSUB7300
950 JH=CM:PRINT:PRINT
960 IF HX=1THEN D=JH:GOSUB5500:JH$=H$:GOTO965
962 JH$=STR$(JH)
965 D=PEEK(JH):GOSUB5500:JM$=H$:GOSUB8000
970 IF VL=0THENMN$="???":BY$="":GOTO1010
990 MN$=M$(VL,1):BY=M(VL)-1:BY$=""
995 IFBY=0THEN1010
1000 FORL=1TOBY:D=PEEK(JH+L):GOSUB5500:BY$=H$+BY$:NEXT
1005 BY$=" $"+BY$:IF CH=4ANDRIGHT$(MN$,1)="▮"THEN1100
1010 PRINTJH$,MN$;BY$
1020 JH=JH+BY+1
1025 IFPEEK(197)=60THEN1025
1026 IFPEEK(197)=1THEN1040
1030 IF JH<=ACTHEN960
1040 PRINT#1:CLOSE1:GOSUB7500
1060 RETURN
1100 REM **IMPRESORA**
1110 L=LEN(MN$):IFRIGHT$(MN$,1)="▮"THENMN$=LEFT$(MN$,L-1):GOTO1110
1120 F=3:IF MID$(MN$,4,1)="("THENF=4
1130 M1$=LEFT$(MN$,F):M2$=RIGHT$(MN$,F-1)
1140 PRINTJH$,M1$;BY$;M2$
1150 GOTO1020
4000 END
4500 REM *****SEPARAR CODIGOS********
```

al Monitor). Para ello hemos de dar el nombre del programa (no es necesario si se trabaja con cinta) y la dirección de inicio de carga (omitir en caso de que se quiera cargar en la misma dirección que estaba al grabarlo).

4. VOLCADO: Nos muestra el contenido de la memoria en valores decimales o hexadecimales.

5; EJECUTAR C.M.: Con esta opción podemos ejecutar (SYS) el programa en C.M. residente en memoria sin más que dar la dirección de llamada.

6. MNEMONICOS: Es quizá la parte más interesante del programa. Permite desensamblar cualquier zona de la memoria. Los códigos mnemónicos están en sentencias DATA, que son leídas al principio del programa.

EXPLICACION DEL PROGRAMA

Cada subrutina está convenientemente indicada por un REM, con lo que su comprensión se hace más fácil; de todos modos paso a explicar las que considero más importantes.

— Las subrutinas de las distintas opciones ocupan de las líneas 10 a 4.000. No las explico paso a paso, pues son lo suficientemente claras.

— Sub. 4500-4900: Se encarga al trabajar con el ensamblador de separar todos los valores que se hayan tecleado, indicar si hay error y POKEarlos en las zonas correspondientes de la memoria. La Sub. 6000-6060 se encarga de hacer lo mismo para los valores decimales.

— Subs. 5000-5060 y 5500-5550: Se encargan de traducir los números de hexadecimal a decimal y decimal a hexadecimal, respectivamente.

— Sub. 7400-7470: Esta rutina realiza las operaciones de LOAD y SAVE de programas en C.M. Para poder cargar y grabar fácilmente los programas he utilizado las rutinas del KERNAL. La primera rutina (SETLFS) se encarga de abrir el periférico. En caso de que se quiera utilizar el disco habría que poner (línea 7405): POKE 781,8 (en vez del POKE 781,1).

La siguiente rutina (SETNAM) controla el nombre del programa; este nombre se coloca en código ASCII a partir de la posición 1024 (memoria de pantalla). Luego se procede a llamar a SAVE o LOAD según la opción escogida, cargando el acumulador y los registros X,Y con los valores apropiados.

Al pasar el programa hay que tener cuidado con las líneas marcadas, pues al sobrepasar los 80 caracteres hay que teclear las instrucciones en su forma abreviada.

EJECUCION DEL PROGRAMA

Al ejecutar el programa se cargan

Programa monitor para el 64

```
4505 ER=0
4510 N1=LEN(V$)
4520 IFLEFT$(V$,1)=" "THENV$=RIGHT$(V$,N1-1):GOTO4510
4525 IF HX=0THEN GOSUB6000:GOTO4590
4530 FORJ=1TON1STEP3
4540 V$(J)=MID$(V$,J,2):IFMID$(V$,J+2,1)<>" "ANDJ<N1-1THEN ER=1:J=N1:GOTO4570
4550 H$=V$(J):GOSUB5000:V(J)=D:GOTO4570
4570 NEXT
4580 CD=INT(N1/3+.5)
4590 IF ER=1THEN 4900
4595 K=1
4600 FORJ=1TOCD
4610 POKE DIR,V(K)
4615 K=K+3:DIR=DIR+1
4620 NEXTJ
4630 IF HX=1THEN D=DIR:GOSUB5500:DIR$=H$:GOTO4900
4640 DIR$=STR$(DIR)
4900 RETURN
5000 REM ***HEX A DEC***************
5010 D=0:N=LEN(H$):IFN<2THEN ER=1:RETURN
5020 FORI=NTO1STEP-1
5030 A$=MID$(H$,I,1):A=ASC(A$):A=A-48:IFA>16THENA=A-7
5035 IFA<0ORA>15THEN ER=1:I=1:GOTO5050
5040 D=D+A*16↑(N-I)
5050 NEXT
5060 RETURN
5500 REM ***DEC A HEX***************
5505 H$=""
5510 C=INT(D/16)
5520 R=D-16*C+1
5530 H$=MID$(HX$,R,1)+H$
5540 IF C<>0THEN D=C:GOTO5510
5545 IFLEN(H$)<2THENH$="0"+H$
5550 RETURN
6000 REM ***SEP. COD. DECIMAL********
6010 J=0:W=1:K=1
6020 J=J+1:IFMID$(V$,J,1)<>" "ANDJ<N1THENGOTO6020
6030 V(W)=VAL(MID$(V$,K,J-K+1)):IFV(W)>255ORV(W)<0THEN ER=1:RETURN
6040 IF J<N1THENW=W+3:K=J:GOTO6020
6050 CD=INT(W/3)+1
6060 RETURN
7000 REM ********HEX?-DEC?**********
7005 HX=0:PRINT"▣▣H▬EXADECIMAL O ▣D▬ECIMAL?"
7010 GETA$:IFA$<>"H"ANDA$<>"D"THEN7010
7020 IFA$="H"THEN HX=1
7030 RETURN
7100 REM***** DIR. INIC - FIN********
7110 PRINT"▣▣DIRECCION INICIO: ";:IF HX=1THEND=CM:GOSUB5500:IN$=H$:GOTO7130
7120 IN$=STR$(CM)
7130 PRINTIN$
7140 PRINT"DIRECCION FINAL: ";:IF HX=1THEN D=AC:GOSUB5500:FIN$=H$:GOTO7160
7150 FIN$=STR$(AC)
7160 PRINTFIN$
7165 IFCM<0ORAC>65535ORAC<CMTHENPRINT"▣ERROR.":GOTO7190
7170 PRINT"▣CORRECTO? (S/N)"
7180 GETA$:IFA$<>"S"ANDA$<>"N"ANDA$<>"M"THENGOTO7180
```

Programa monitor para

```
7185 IFA$="S"ORA$="M"THEN7220
7190 INPUT"INICIO";CM$:INPUT"FINAL";AC$:IFHX=0THENCM=VAL(CM$):AC=VAL(AC$):GOTO72
10
7200 H$=CM$:GOSUB5000:CM=D:H$=AC$:GOSUB5000:AC=D
7210 GOTO7110
7220 RETURN
7300 REM******SAL. IMPRESORA*********
7310 PRINT"SALIDA IMPRESORA? (S/N)"
7320 GETA$:IFA$="S"THEN CH=4:GOTO7340
7325 IFA$="N"THENCH=3:GOTO7340
7330 GOTO7320
7340 OPEN1,CH:CMD1
7350 RETURN
7400 REM***LOAD/SAVE*****************
7402 PRINT"▣":SA=0:ERR=0:IF LO=0OR(LO=1ANDDC=0)THENSA=1
7405 POKE780,1:POKE781,1:POKE782,SA:SYS(65466):REM SETLFS
7410 POKE780,L:POKE781,0:POKE782,4:SYS(65469):REM SETNAM ($0400-NOMBRE)
7415 IF LO=1THEN7440
7420 POKE251,CM-256*INT(CM/256):POKE252,INT(CM/256):REM PAG 0. INIC.GRAB.
7430 POKE780,251:POKE781,AC-256*INT(AC/256):POKE782,INT(AC/256):SYS(65496)
7435 GOTO7460
7440 C1=0:C2=0:IFSA=0THENC2=INT(DC/256):C1=DC-256*C2
7450 POKE780,0:POKE781,C1:POKE782,C2:SYS65493
7460 IFPEEK(780)<>0 OR ST<>0 THENPRINT"ERROR!!!!":FORI=1TO3500:NEXT:ERR=1
7470 PRINT"▣":RETURN
7500 REM ********PAUSA***************
7510 PRINT"▣        PULSA 'C' PARA CONTINUAR"
7520 GETA$:IFA$<>"C"THEN7520
7530 RETURN
8000 REM***ERROR MNEMO?**************
8005 VL=0
8010 FORX=1TO151
8020 VL=-X*(JM$=M$(X,2)):IFVLTHENX=151
8030 NEXT
8040 RETURN
8200 REM***A$-ASCII*****************
8202 L=LEN(A$)
8205 IFA$=""THEN8230
8210 FORJ=1TOL:A=ASC(MID$(A$,J,1)):POKE1023+J,A:POKE55295+J,6:NEXT
8230 PRINT"▣▣▣▣▣";:RETURN
8500 FORI=1TO151
8510 READ M$(I,1),M$(I,2),M(I)
8520 NEXT
8530 RETURN
8999 REM*********DATAS*****************
9000 DATA ADC#,69,2,ADC,65,2,"ADC      ,X██████I",75,2,ADC,6D,3
9002 DATA "ADC        ,X████████I",7D,3,"ADC        ,Y████████I",79,3
9004 DATA"ADC(      ,X)████████I",61,2,"ADC(      ),Y███████I",71,2
9006 DATA AND#,29,2,AND,25,2,"AND      ,X██████I",35,2,AND,2D,3
9008 DATA "AND        ,X████████I",3D,3,"AND        ,Y████████I",39,3
9010 DATA "AND(      ,X)████████I",21,2,"AND(      ),Y███████I",31,2
9012 DATAASL A,0A,1,ASL,06,2,"ASL      ,X██████I",16,2,ASL,0E,3
9014 DATA "ASL        ,X████████I",1E,3,BCC,90,2,BCS,B0,2,BEQ,F0,2,BIT,24,2
9016 DATA BIT,2C,3,BMI,30,2,BNE,D0,2,BPL,10,2,BRK,00,1,BVC,50,2,BVS,70,2
9018 DATA CLC,18,1,CLD,D8,1,CLI,58,1,CLV,B8,1,CMP#,C9,2,CMP,C5,2
9020 DATA"CMP      ,X██████I",D5,2,CMP,CD,3,"CMP        ,X████████I",DD,3
```

el 64

```
9022 DATA"CMP       ,Y▮▮▮▮▮▮▮▮▮",D9,3,"CMP(    ,X)▮▮▮▮▮▮▮▮",C1,2
9024 DATA"CMP(    ),Y▮▮▮▮▮▮▮▮",D1,2,CPX#,E0,2,CPX,E4,2,CPX,EC,3
9026 DATA CPY#,C0,2,CPY,C4,2,CPY,CC,3,DEC,C6,2,"DEC    ,X▮▮▮▮▮▮▮",D6,2
9028 DATA DEC,CE,3,"DEC       ,X▮▮▮▮▮▮▮▮▮",DE,3,DEX,CA,1,DEY,88,1
9030 DATA EOR#,49,2,EOR,45,2,"EOR    ,X▮▮▮▮▮▮▮",55,2,EOR,4D,3
9032 DATA "EOR       ,X▮▮▮▮▮▮▮▮▮",5D,3,"EOR       ,Y▮▮▮▮▮▮▮▮▮",59,3
9034 DATA "EOR(    ,X)▮▮▮▮▮▮▮▮",41,2,"EOR(    ),Y▮▮▮▮▮▮▮▮",51,2
9036 DATAINC,E6,2,"INC    ,X▮▮▮▮▮▮▮",F6,2,INC,EE,3,"INC       ,X▮▮▮▮▮▮▮▮▮",FE,3
9038 DATA INX,E8,1,INY,C8,1,JMP,4C,3,"JMP(       )▮▮▮▮▮▮▮▮",6C,3,JSR,20,3
9040 DATALDA#,A9,2,LDA,A5,2,"LDA    ,X▮▮▮▮▮▮▮",B5,2,LDA,AD,3,"LDA       ,X▮▮▮▮▮▮▮▮"
9042 DATA BD,3,"LDA       ,Y▮▮▮▮▮▮▮▮▮",B9,3,"LDA(    ,X)▮▮▮▮▮▮▮▮",A1,2
9044 DATA "LDA(    ),Y▮▮▮▮▮▮▮▮",B1,2,LDX#,A2,2,LDX,A6,2,"LDX    ,Y▮▮▮▮▮▮▮",B6,2
9046 DATALDX,AE,3,"LDX       ,Y▮▮▮▮▮▮▮▮▮",BE,3
9048 DATA LDY#,A0,2,LDY,A4,2,"LDY    ,X▮▮▮▮▮▮▮",B4,2,LDY,AC,3
9050 DATA "LDY       ,X▮▮▮▮▮▮▮▮▮",BC,3,LSR A,4A,1,LSR,46,2,"LSR    ,X▮▮▮▮▮▮▮",56,2
9052 DATA LSR,4E,3,"LSR       ,X▮▮▮▮▮▮▮▮▮",5E,3
9054 DATANOP,EA,1,ORA#,09,2,ORA,05,2,"ORA    ,X▮▮▮▮▮▮▮",15,2,ORA,0D,3
9056 DATA"ORA       ,X▮▮▮▮▮▮▮▮▮",1D,3,"ORA       ,Y▮▮▮▮▮▮▮▮▮",19,3,"ORA(    ,X)▮▮▮▮▮
▮▮▮"
9058 DATA 01,2,"ORA(    ),Y▮▮▮▮▮▮▮▮",11,2,PHA,48,1,PHP,08,1,PLA,68,1,PLP,28,1
9060 DATAROL A,2A,1,ROL,26,2,"ROL    ,X▮▮▮▮▮▮▮",36,2,ROL,2E,3,"ROL       ,X▮▮▮▮▮▮▮
▮▮"
9061 DATA 3E,3,ROR A
9062 DATA6A,1,ROR,66,2,"ROR    ,X▮▮▮▮▮▮▮",76,2,ROR,6E,3,"ROR       ,X▮▮▮▮▮▮▮▮▮
9063 DATA7E,3
9064 DATARTI,40,1,RTS,60,1,SBC#,E9,2,SBC,E5,2,"SBC    ,X▮▮▮▮▮▮▮",F5,2,SBC,ED,3
9066 DATA"SBC    ,X▮▮▮▮▮▮▮",FD,3,"SBC       ,Y▮▮▮▮▮▮▮▮▮",F9,3,"SBC(    ,X)▮▮▮▮▮▮▮▮
"
9068 DATA E1,2,"SBC(    ),Y▮▮▮▮▮▮▮▮",F1,2,SEC,38,1,SED,F8,1,SEI,78,1,STA,85,2
9070 DATA"STA    ,X▮▮▮▮▮▮▮",95,2,STA,8D,3,"STA       ,X▮▮▮▮▮▮▮▮▮",9D,3
9072 DATA"STA       ,Y▮▮▮▮▮▮▮▮▮",99,3,"STA(    ,X)▮▮▮▮▮▮▮▮",81,2,"STA(    ),Y▮▮▮▮▮▮
▮▮"
9074 DATA 91,2,STX,86,2,"STX    ,Y▮▮▮▮▮▮▮",96,2,STX,8E,3,STY,84,2
9076 DATA"STY    ,X▮▮▮▮▮▮▮",94,2,STY,8C,3,TAX,AA,1,TAY,A8,1,TSX,BA,1
9078 DATA TXA,8A,1,TXS,9A,1,TYA,98,1
```

```
DIR. INICIO?
? C000
```

```
C000? A9 04
C002? 20 C0 FF
C005? M
```

los códigos mnemónicos. Antes de empezar a trabajar con cualquier opción habrá que indicar si trabajaremos en hexadecimal o decimal; a continuación el ordenador nos dará, en caso necesario, las direcciones de inicio y fin que corresponden a las posiciones primera y última al trabajar en ENSAMBLADOR (que serán cero en caso de no haber hecho ningún programa). En las opciones de volcado en pantalla se permite realizar dicho vuelco en una impresora preguntándose la opción escogida. En todos los casos se vuelve al menú par abortar la opción, pulsando la "M".

José Luis de Prat

Concurso

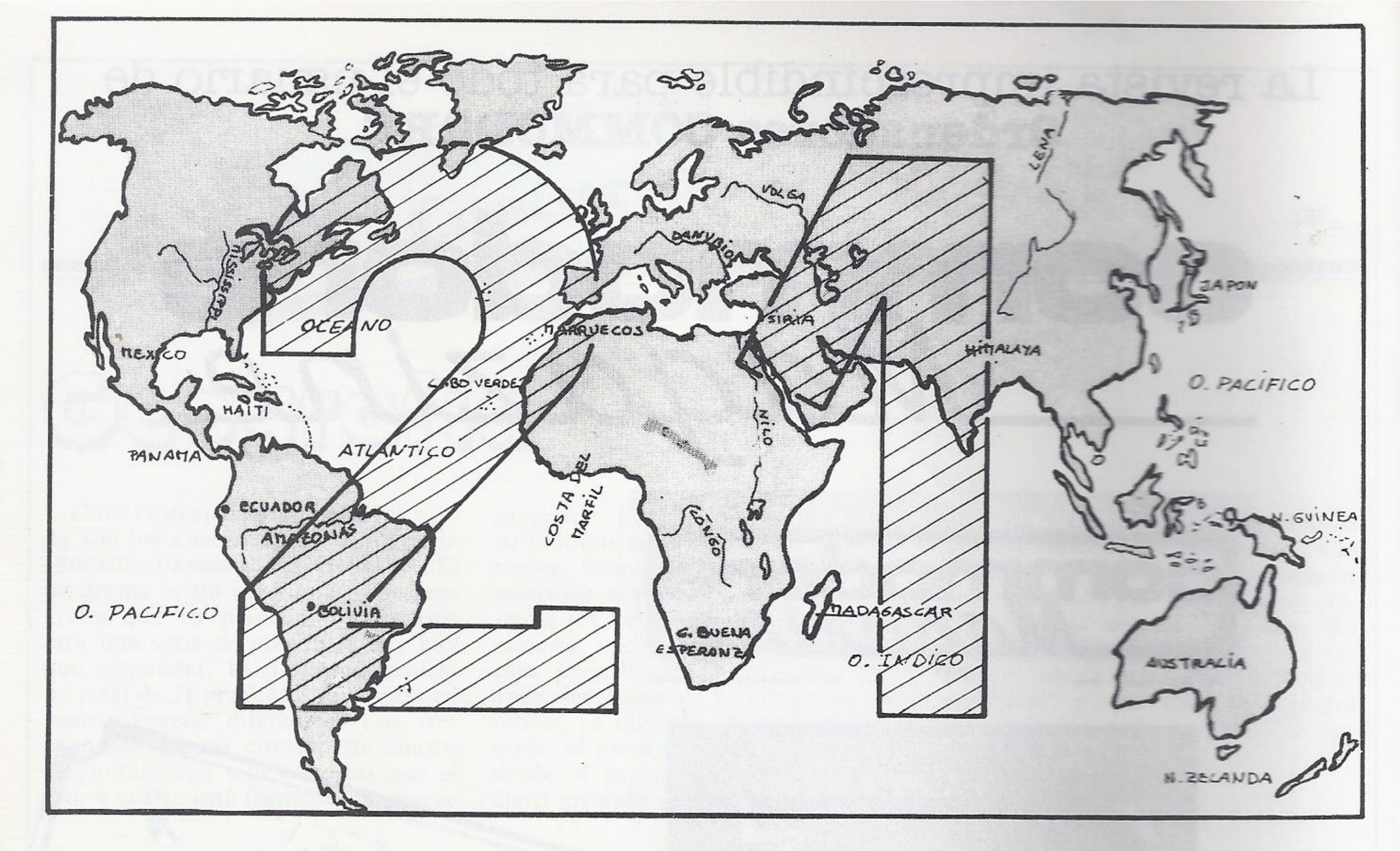
Geografia

C-64

Dani Horcajada y Rubén Horcajada son los autores de este programa educativo llamado GEOGRAFIA. El programa es un cuestionario de geografía que va planteando una tras otra una serie de preguntas que hay que responder. El programa realiza un total de 21 preguntas, divididas en cuatro grupos diferentes. Los tres primeros grupos constan de cuatro preguntas cada uno mientras que el grupo cuatro está formado por nueve preguntas. El programa es un ejemplo de la importante ayuda que un ordenador, bien programado, puede suministrar tanto a los estudiantes como a los "enseñantes" de cualquier materia. Además, la estructura de este progrma lo hace ideal para aprender listas de nombres de memoria, ya que cuando se comete un error, el programa obliga a empezar desde el principio. También es un claro ejemplo de cómo el color, los gráficos y el sonido pueden ser una útil herramienta que haga mucho más llevadero el tener que aprenderse algún "rollo". Por último, cabe destacar que la sencilla estructura del programa hace de él, como dicen sus autores, un progrma abierto, fácil de modificar, y que cada uno podrá adaptar con toda sencillez a sus necesidades particulares.

```
10 PRINT"▒"
20 POKE53280,0:POKE53281,0
21 PRINT"[illegible]"
22 PRINT"[illegible]"
23 PRINT"[illegible]"
24 PRINT"▒♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦"
25 PRINT"▒PROGRAMA IDEADO Y REALIZADO POR:"
26 PRINTTAB(23)"▒▒DANI HORCAJADA"
27 PRINT"▒▒GRAFICOS,COLOR Y SONIDO POR:"
28 PRINTTAB(23)"▒▒RUBEN HORCAJADA"
29 PRINT"▒♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦"
35 PRINTTAB(10)"▒PEQUENAS INSTRUCCIONES"
36 PRINT"▒▒TODO CONSISTE EN ADIVINAR EL NOMBRE"
37 PRINT"CORRECTO  DE LA CAPITAL DEL PAIS"
38 PRINT"QUE SE OS PIDE..."
39 PRINT"▒SUERTE!!!...Y AL LORO!!!"
40 VV=212*256
41 POKEVV+24,15
42 POKEVV+5, 9+16
43 POKEVV+6,0
44 REM BUCLE PRINCIPAL
45 READ F,N
46 IF F=0 THENPOKE VV+24,0:GOTO 84
47 FF=16*F
48 POKE VV+1,FF/256
50 POKEVV,(FF-32768)AND 255
51 T=TI+18*N
52 POKEVV+4,17
53 IF TI<T THEN53
54 POKE VV+4,0
55 GOTO 44
56 DATA 287,2
57 DATA 384,2
58 DATA 384,2
59 DATA 384,2
60 DATA 384,1
61 DATA 483,3
```

PREMIADO CON 5.000 PESETAS




```
62 DATA 384,1
63 DATA 384,1
64 DATA 431,2
65 DATA 431,2
66 DATA 431,2
67 DATA 431,1
68 DATA 512,3
69 DATA 431,1
70 DATA 483,3
71 DATA 431,1
72 DATA 384,7
73 DATA 0,0
84 PRINT"♥":POKE53281,0:POKE53280,6
85 PRINTTAB(8)"▓▓▓▓▓▓▓▓▓▓CUAL ES LA CAPITAL DE...?"
86 PRINTTAB(7)"▓+++++++++++++++++++++++++++++++"
87 PRINT"▓▓RECUERDA:▓PULSA RETURN DESPUES DE CADA"
88 PRINT"RESPUESTA.        ▓────────"
93 PRINT"▓▓▓...ALBANIA?"
94 INPUT A$
95 IF A$="TIRANA"THEN PRINT"▓CORRECTO"
96 IF A$<>"TIRANA"THEN PRINT "▓INCORRECTO,ES TIRANA":GOTO10000
99 PRINT"▓▓▓...YUGOSLAVIA?"
100 INPUT B$
110 IF B$="BELGRADO"THEN PRINT "▓CORRECTO"
120 IF B$<>"BELGRADO"THEN PRINT "▓INCORRECTO,ES BELGRADO":GOTO10000
130 PRINT"▓▓▓...REP.SUDAFRICANA?"
140 INPUT C$
150 IF C$="CIUDAD DE EL CABO" THEN PRINT"▓CORRECTO"
160 IFC$<>"CIUDAD DE EL CABO"THEN PRINT"▓INCORRECTO,ES CIUDAD DE EL CABO":GOTO10
000
170 PRINT"▓▓▓...GRECIA?"
180 INPUT D$
190 IF D$="ATENAS" THEN PRINT"▓CORRECTO"
200 IF D$<>"ATENAS"THEN PRINT "▓INCORRECTO,ES ATENAS":GOTO10000
205 PRINT"▓▓▓SIGO PREGUNTANDO."
206 PRINT"▓▓CUAL ES LA CAPITAL DE...?"
210 PRINT"▓▓▓...ALEMANIA OCC.?"
```

```
220 INPUT E$
230 IF E$="BONN" THEN PRINT"◣CORRECTO"
240 IF E$<>"BONN"THEN PRINT"▒INCORRECTO,ES BONN":GOTO10000
250 PRINT"▩▩▩...HUNGRIA?"
260 INPUT F$
270 IF F$="BUDAPEST" THEN PRINT"◣CORRECTO"
280 IF F$<>"BUDAPEST"THEN PRINT"▒INCORRECTO,ES BUDAPEST":GOTO10000
290 PRINT"▩▩▩...CAMERUN?"
300 INPUT G$
310 IF G$="YAOUNDE" THEN PRINT"◣CORRECTO"
320 IF G$<>"YAOUNDE"THEN PRINT"▒INCORRECTO,ES YAOUNDE":GOTO10000
330 PRINT"▩▩▩...NUEVA ZELANDA?"
340 INPUT H$
350 IF H$="WELLINGTON"THEN PRINT"◣CORRECTO"
360 IF H$<>"WELLINGTON"THEN PRINT"▒INCORRECTO,ES WELLINGTON":GOTO10000
370 PRINT "▩▩▩COMO VEREIS YO NO ME CANSO Y..."
380 PRINT"▩SIGO PREGUNTANDO!!!"
390 PRINT"▩▩CUAL ES LA CAPITAL DE...?"
400 PRINT"▩▩▩...ISLANDIA?"
410 INPUT I$
420 IF I$="REYKJAVIK"THEN PRINT"◣CORRECTO"
430 IF I$<>"REYKJAVIK"THEN PRINT"▒INCORRECTO,ES REYKJAVIK":GOTO10000
440 PRINT"▩▩▩...AFGANISTAN?"
450 INPUTF$
460 IF F$="KABUL"THEN PRINT"◣CORRECTO"
470 IF F$<>"KABUL"THEN PRINT"▒INCORRECTO,ES KABUL":GOTO10000
480 PRINT"▩▩▩...HONDURAS?"
490 INPUT G$
500 IF G$="TEGUCIGALPA"THEN PRINT"◣CORRECTO"
510 IF G$<>"TEGUCIGALPA"THEN PRINT"▒INCORRECTO,ES TEGUCIGALPA":GOTO10000
520 PRINT"▩▩▩...AUSTRALIA?"
530 INPUT H$
540 IF H$="CANBERRA"THEN PRINT"◣CORRECTO"
550 IF H$<>"CANBERRA"THEN PRINT"▒INCORRECTO,ES CANBERRA":GOTO10000
600 PRINT"▩▩▩SI HABEIS SUPERADO YA LOS TRES APARTA-"
610 PRINT"DOS ANTERIORES,ADELANTE!!!."
620 PRINT"AQUI TENEIS EL SIGUIENTE,UN POCO MAS "
630 PRINT"DIFICILILLO QUE LOS OTROS TRES."
640 PRINT"▩TU SOLO TIENES QUE CONTESTAR CORRECTA-"
650 PRINT"MENTE A LAS PREGUNTAS QUE TE FORMULAMOS"
660 PRINTTAB(16)"▩▩SUERTE!!!"
665 FORX=0TO11000:NEXT
670 PRINT"▩▩▩CUAL ES LA MAYOR NACION DEL MUNDO?"
680 INPUTM$
690 IFM$="LA URSS"THEN PRINT"◣CORRECTO"
700 IFM$<>"LA URSS"THEN PRINT"▒INCORRECTO,ES LA URSS":GOTO10000
780 PRINT"▩▩▩CUAL ES EL PAIS MAS POBLADO DEL MUNDO?"
790 INPUTO$
800 IFO$="CHINA"THEN PRINT"◣CORRECTO"
810 IFO$<>"CHINA"THEN PRINT"▒INCORRECTO,ES CHINA":GOTO10000
820 PRINT"▩▩▩CUAL ES EL TERRITORIO MAS DENSAMENTE "
830 PRINT"POBLADO DEL MUNDO?"
840 INPUTP$
850 IFP$="MACAO"THEN PRINT"◣CORRECTO"
860 IFP$<>"MACAO"THEN PRINT"▒INCORRECTO,ES MACAO":GOTO10000
```

Concurso

Viene de la página anterior

```
870 PRINT"CUAL ES EL TERRITORIO MENOS DENSAMENTE"
880 PRINT"POBLADO DEL MUNDO?"
890 INPUTQ$
900 IFQ$="LA ANTARTIDA"THEN PRINT"CORRECTO"
910 IFQ$<>"LA ANTARTIDA"THENPRINT"INCORRECTO,ES LA ANTARTIDA":GOTO10000
920 PRINT"CUAL ES EL PAIS CON MAS INDICE DE EMI-"
930 PRINT"GRACION?"
940 INPUTR$
950 IFR$="MEJICO"THEN PRINT"CORRECTO"
960 IFR$<>"MEJICO"THENPRINT"INCORRECTO,ES MEJICO":GOTO10000
970 PRINT"CUAL ES LA PENINSULA MAS EXTENSA DEL"
980 PRINT"MUNDO?"
990 INPUTS$
1000 IFS$="ARABIA"THEN PRINT"CORRECTO"
1010 IFS$<>"ARABIA"THENPRINT"INCORRECTO,ES ARABIA":GOTO10000
1020 PRINT"CUAL ES EL DESIERTO MAS GRANDE DE EL"
1030 PRINT"MUNDO?"
1040 INPUTT$
1050 IFT$="SAHARA"THENPRINT"CORRECTO"
1060 IFT$<>"SAHARA"THENPRINT"INCORRECTO,ES SAHARA":GOTO10000
1070 PRINT"CUAL ES LA CIUDAD AMURALLADA MAS AN-"
1080 PRINT"TIGUA DEL MUNDO?"
1090 INPUTU$
1100 IFU$="JERICO"THENPRINT"CORRECTO"
1110 IFU$<>"JERICO"THENPRINT"INCORRECTO,ES JERICO":GOTO10000
1120 PRINT"Y PARA FINALIZAR,OS HARE UNA PREGUN-"
1130 PRINT"TA DE LO MAS ''TIRAO'':"
1140 FORX=0TO3000:NEXT
1150 PRINT"CUAL ES LA CIUDAD MAS HABITADA DEL MUN-"
1160 PRINT"DO?"
1170 INPUTV$
1180 IFV$="TOKIO"THENPRINT"CORRECTO"
1190 IFV$<>"TOKIO"THENPRINT"INCORRECTO,ES TOKIO":GOTO10000
1200 PRINT"QUE TAL?.DIFICIL?.NO,HOMBRE NO,QUE ESTO"
1210 PRINT"NO ES TAN COMPLICADO.OK?"
1220 PRINT"COMO VEIS AQUI SE ACABA EL GRANDIOSO"
1230 PRINT"''MACHAQUEO''DE PREGUNTAS."
1240 PRINT"SUPONGO QUE AL FINAL OS SABREIS LAS 21?"
1250 PRINTTAB(15)"HASTA OTRA!!!"
1260 PRINTTAB(15)"————————————"
1300 FORX=0TO9000:NEXT
2000 PRINT""
2010 FORX1=0TO22
2020 C1$=""
2030 GOSUB2090
2040 FORT=1TO150:NEXT
2050 C1$=""
2070 NEXT
2080 GOTO2190
2090 REM AVION
2100 PRINT""C1$
2105 IFX1=0THEN2120
2110 FORJJ=1TOX1:PRINT"";:NEXT
```

```
2120 I1$="▛▚▚▚▚▚ADIOS!!!"
2130 D1$="▔▏"
2140 E1$="▬▬▬ ◠"
2150 F1$="▚▔▔▚▀▀"
2160 G1$=I1$+D1$+E1$+F1$
2170 PRINTG1$;
2180 RETURN
2190 PRINT"▚▚    COPYRIGHT (C) DANI HORCAJADA 1984"
2200 PRINTTAB(15)"▚& RUBEN"
9999 END
10000 V=54296:W=54276:A=54277:H=54273:L=54272
10001 FORX=15TO0STEP-1:POKEV,X:POKEW,129:POKEA,15:POKEH,40:POKEL,200:NEXT
10002 POKEW,0:POKEA,0
10003 PRINT"▚▚▚▚COMO TE HAS EQUIVOCADO"
10004 PRINT"EL JUEGO SE HA ACABADO"
10005 PRINT"▚▚PERO SI QUIERES VOLVER A EMPEZAR..."
10006 PRINT"ESCRIBE LA PALABRA RUN    ▚▔▔▔▔▔▔"
10007 PRINTTAB(19)"▚▔▔▔"
```





PREMIADO CON 5.000 PESETAS

Los Sprites son una de las más interesantes posibilidades gráficas con que cuenta el **C-64**, pero para diseñarlos se requiere una buena dosis de trabajo. Para ayudar en esta labor, Juan Irisarri, de Madrid, ha enviado a concursar este programa, SPRITES, que es un auténtico editor de Sprites con bastantes posibilidades. El programa se maneja de forma sencilla desde un menú que ofrece las siguientes opciones:

1. **CREAR SPRITE.** Esta opción permite diseñar por primera vez un Sprite. Esto se lleva a cabo sobre una rejilla, introduciendo las coordenadas de los puntos que se desea encender.
2. **BORRAR PIXEL.** Esta segunda opción ofrece la posibilidad de ir borrando uno a uno los pixels encendidos que no interesen.
3. **VISIONAR.** Al elegir esta opción desaparece el reticulado de la pantalla, lo que permite ver el Sprite, tal y como está siendo diseñado, en todo su esplendor, sin líneas de por medio.
4. **CODIGOS.** Aparecen, línea por línea, los valores codificados correspondientes al Sprite.
5. **REPRESENTAR.** Permite ver el Sprite, tal y como aparecerá en un programa, de varias formas. Estas formas incluyen el aspecto normal del Sprite, el mismo expandido verticalmente, expandido horizontalmente y por último, expandido en ambas direcciones, tanto horizontal como verticalmente.

6 FIN. Opción que, para los que no lo hayan adivinado, permite terminar el programa.

Felicitamos a Juan, el autor de este programa, con el que no se nos va a resistir ni un solo Sprite.




```
10 REM*** PANTALLA Y MENU ***
20 B=1024:Y1=54272
30 POKE53281,14:POKE646,2
32 A$(0)="   OPCIONES    ":A$(1)="1-CREAR SPRITE":A$(2)="2-BORRAR PIXEL"
33 A$(3)="3-VISIONAR     ":A$(4)="4-CODIGOS      ":A$(5)="5-REPRESENTAR "
34 A$(6)="0-FIN          "
39 PRINT"□"
40 FORI=1TO21:POKEB+I,I:POKEB+I+Y1,0:POKEB+40*I,I:POKEB+40*I+Y1,0:NEXT
50 FORI=22TO24:POKEB+I,I:POKEB+I+Y1,0:NEXT
60 FORI=0TO20:FORJ=1065TO1088:IFJ=1072ORJ=1080ORJ=1088THENCL=5:GOTO80
70 CL=6
80 POKEJ+40*I,91:POKEJ+40*I+Y1,CL:NEXT:NEXT
90 X=26:Y=5:GOSUB5100:PRINT"■"A$(0):JL=1
95 GETJD$:IFJD$>""THEN95
100 FORY=7TO17STEP2:GOSUB5100:PRINTA$(JL):JL=JL+1:NEXT
110 X=10:Y=22:GOSUB5100:INPUT"OPCION";O
120 IFO<0ORO>5THEN90
130 ONO+1GOTO200,500,500,600,800,1000
200 PRINT"□□";:END
500 GOSUB5000:X=26:Y=5:GOSUB5100:PRINT"■"A$(O)
510 Y=7:GOSUB5100:PRINT"  INTRODUCE ":Y=9:GOSUB5100:PRINT"COORDENADAS"
520 Y=11:GOSUB5100:PRINT" (EJEMPLO AX)":Y=17:GOSUB5100:PRINT"   (0=FIN) "
530 X=30:Y=15:GOSUB5100:GETP$:PRINTP$:IFP$=""THEN530
531 IFP$="0"THEN90
535 X=31:GOSUB5100:GETPP$:PRINTPP$:IFPP$=""THEN535
540 GOSUB5020:X1=ASC(P$)-65:Y2=ASC(PP$)-65:P=1065+Y2+X1*40
550 IFO=1THENPS=160:C1=0:GOTO580
560 PS=91:IFY2=7ORY2=15ORY2=23THENC1=5:GOTO580
570 C1=6
580 POKEP,PS:POKEP+Y1,C1
590 X=26:Y=15:GOSUB5100:PRINT"   ":GOTO530
600 GOSUB5000:X=26:Y=5:GOSUB5100:PRINT"■"A$(3)
610 FORI=0TO20:FORJ=1065TO1088
620 IFPEEK(J+40*I)<>91THEN640
630 POKEJ+40*I,32:POKEJ+40*I+Y1,14
640 NEXT:NEXT
650 Y=11:GOSUB5100:PRINT"PULSA ESPACIO"
660 Y=13:GOSUB5100:PRINT"  PARA MENU  "
670 IFPEEK(197)<>60THEN670
680 FORI=0TO20:FORJ=1065TO1088
690 IFPEEK(J+40*I)=160THEN730
700 IFJ=1072ORJ=1080ORJ=1088THENC1=5:GOTO720
710 C1=6
720 POKEJ+40*I,91:POKEJ+40*I+Y1,C1
730 NEXT:NEXT
740 GOSUB5000:GOTO90
800 GOSUB5000:X=26:Y=0:GOSUB5100:PRINT"■"A$(4)
810 FORI=0TO20
820 FORJ=0TO2:CD%=0
830 FORK=0TO7
840 R=PEEK(1065+J*8+K+I*40):IFR=91THEN860
```

```
850 CD%=CD%+2↑(7-K)
860 NEXTK
870 IFJ=2THENCD$=STR$(CD%):GOSUB990:GOTO890
880 CD$=STR$(CD%):GOSUB990:CD$=CD$+","
890 X=26+5*J:Y=I+1:GOSUB5100:PRINTCD$
900 NEXTJ
920 NEXTI
930 X=10:Y=22:GOSUB5100:PRINT"▒PULSE ESPACIO PARA MENU"
940 IFPEEK(197)<>60THEN940
955 FORY=0TO21
960 X=26:GOSUB5100:PRINT"              "
965 NEXT
970 X=10:Y=22:GOSUB5100:PRINT"                          "
980 GOTO90
990 IFLEN(CD$)=1THENCD$=CD$+"   "
991 IFLEN(CD$)=2THENCD$=CD$+"  "
992 IFLEN(CD$)=3THENCD$=CD$+" "
993 RETURN
1000 GOSUB5000:X=26:Y=5:GOSUB5100:PRINT"▒"A$(5)
1005 V=53248:N=0
1008 Y=9:GOSUB5100:PRINT"  ESPERE"
1010 FORI=0TO20
1020 FORJ=0TO2:CD%=0
1030 FORK=0TO7
1040 R=PEEK(1065+J*8+K+I*40):IFR=91THEN1060
1050 CD%=CD%+2↑(7-K)
1060 NEXTK
1070 POKE832+N,CD%:N=N+1
1080 NEXTJ
1090 NEXTI
1100 POKEV+4,240:POKEV+5,175:POKE2042,13:POKEV+41,0
1102 POKEV+2,240:POKEV+3,100:POKE2041,13:POKEV+40,0
1104 POKEV,240:POKEV+1,150:POKE2040,13:POKEV+39,0
1106 POKEV+6,255:POKEV+7,60:POKE2043,13:POKEV+42,0
1110 POKEV+21,15:POKEV+23,6:POKEV+29,5
1115 X=26:Y=9:GOSUB5100:PRINT"          "
1120 X=10:Y=22:GOSUB5100:PRINT"▒PULSE ESPACIO PARA MENU"
1130 IFPEEK(197)<>60THEN1130
1140 POKEV+21,0
1150 GOSUB5100:PRINT"          "
1160 GOTO90
5000 X=26:Y=5:GOSUB5100:PRINT"          "
5010 FORY=7TO17STEP2:GOSUB5100:PRINT"            ":NEXT:RETURN
5020 REM
5030 IFASC(P$)<65ORASC(P$)>85THEN5050
5040 IFASC(PP$)>64ANDASC(PP$)<89THENRETURN
5050 X=26:Y=15:GOSUB5100:PRINT"      "
5060 X=30:GOSUB5100:GETP$:PRINTP$:IFP$=""THEN5060
5070 X=31:GOSUB5100:GETPP$:PRINTPP$:IFPP$=""THEN5070
5080 GOTO5020
5100 POKE781,Y:POKE782,X:POKE783,0:SYS65520:RETURN
```

PROGRAMA commodore Magazine COMPROBADO

SOFTWARE CENTER

sinclair
SEIKOSHA
indescomp
SHARP
commodore

★LAS MEJORES MARCAS
★EL MEJOR SERVICIO
★LOS PRECIOS MAS JUSTOS
★LA MAYOR GARANTIA



Y AHORA, ADEMAS...

ORIC & SPECTRAVIDEO™

➘ **VIDEO-GAMES**
- CLUB DE VIDEO JUEGOS

➘ **GAMES CLUB**
- CLUB DE USUARIOS, COMMODORE-64, SPECTRUM Y ORIC

Games Club's



RED NACIONAL DE CLUBS

Aceptamos nuevos grupos federados.
Inmejorables condiciones y asesoramiento.

CONSULTENOS !!!

«CONDICIONES ESPECIALES» a Centros de enseñanza, alumnos de informática y clubs de usuarios.

PONEMOS A SU DISPOSICION «EL CATALOGO» DE SOFTWARE MAS Y MEJOR SURTIDO DEL MERCADO PARA *SPECTRUM, COMMODORE* Y *ORIC.*

EL CATALOGO

oferta de verano, solo hasta el 1 de octubre

SHINWA CP80 F/T

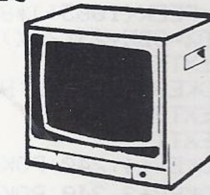
Impresora matricial 80 columnas con set de caracteres españoles, totalmente compatible.

SHINWA CP80 F/T es la nueva impresora. Con tecnología actual y precio competitivo, ofrece las dos características que hoy día hay que exigir a una buena impresora: fiabilidad y calidad de impresión.
Pero la SHINWA CP80 F/T no se queda ahí: ofrece una resolución de 640 puntos por línea, juego de caracteres españoles y una gran variedad de posibilidades en la impresión de textos: normal, comprimido, doble ancho, super índices subíndices reducidos, etc. La impresora se suministra con interface tipo CENTRONICS. Opcionalmente, se puede conectar un interface RS-232

P.V.OFERTA
55 200-

DATALEC

P.V.OFERTA
26 320-



Monitor monocromo para visualización de datos.
El monitor DATALEC, con su pantalla de fósforo verde P-31 de 12 pulgadas, es la pantalla de visualización ideal para presentación de datos y gráficos en alta resolución.

A+L centro de formación. Informática general y BASIC. Prácticas con COMMODORE 64. INFORMACION: c/ Manso, 17 tel.: 325 87 71

SE BUSCAN! los mejores PROGRAMADORES. Pagamos excelentes royalties. Garantia y seriedad total.

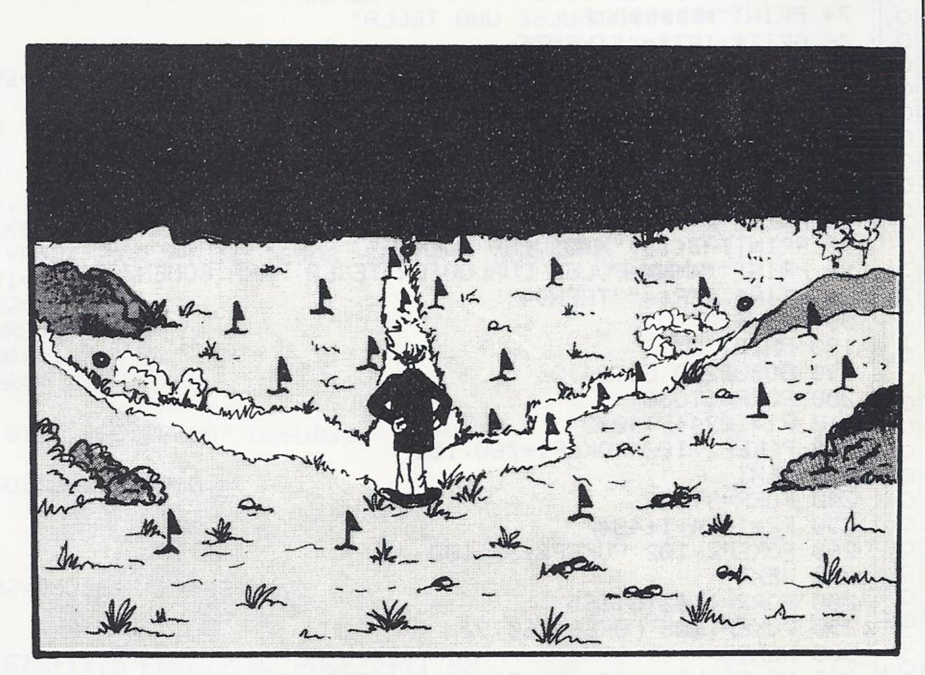
Nombre .... Dirección ....
Población .... Provincia ....
Distrito Postal .... Teléfono ....

BOLETIN DE INFORMACION Tel.: 219 10 90
remitir a
SOFTWARE CENTER
Avda. Mistral 10, 1 D izq. BARCELONA
08015

ATENCION!!! A PARTIR DEL 1° DE AGOSTO ENTRA EN FUNCIONAMIENTO EL «MERCA-TELEFONO» MARCANDO EL *219 10 90 (93)* PODRA COMPRAR-CAMBIAR-VENDER CUALQUIER ARTICULO.

Campo minado

CAMPO MINADO es una versión para el C-64 del conocido juego de "cruce usted el campo de minas llevando un detector que le dice cuántas minas tiene alrededor", juego que se ha convertido ya en un clásico. El autor de esta versión es Jaime Corbella que envía su programa a competir en el concurso desde Barcelona. El programa incluye todas las instrucciones y explicaciones necesarias páara poder jugar, que se pueden leer en la pantalla del televisor un instante después de escribir la famosa palabra *RUN*. En esencia el juego consiste en cruzar de un extremo a otro un campo lleno de minas. La única ayuda que tenemos es un indicador, en la parte superior de la pantalla, que nos dice en todo momento cuántas minas tenemos a nuestro alrededor. el juego se hace a ocho partidas, en cada una de las cuales el número va aumentando, o sea, que cada vez es más probable saltar por los aires simplemente por un falso movimiento del dedo sobre la tecla equivocada. Al acabar una pantalla, tanto si la acabas vivo como a trocitos, el ordenador te mostrará la repetición de la jugada, pero ahora mostrándote donde estaban escondidas las queridas minas con lo que podrás ver las tonterías que pueden llegar a hacerse sobre la pantalla.



Incluimos una pequeña explicación de la estructura del juego que nos envía su autor:

10-96 Explicación y reglas del juego.
140-420 Presentación del campo.
440-500 Colocación aleatoria de las minas.
520-740 Mecanismo del juego.
880-940 Repetición de la jugada.

```
10 DIMX(500):DIMY(500):DIML(1000)
20 PRINT"■"
30 POKE53280,2:POKE53281,1
40 PRINTTAB(12)"■■■ CAMPO MINADO *"
42 PRINT"■■■■■ JAIME CORBELLA  BARCELONA-1984"
44 PRINT"■----------------------------------------"
46 PRINT"EL JUEGO CONSISTE EN HACER LLEGAR LA"
48 PRINT"FIGURA (■♠)■ DE LA PARTE INFERIOR A LA"
50 PRINT"PARTE SUPERIOR DONDE ESTA MARCADO CON"
52 PRINT"LA FIGURA ■↑↑↑■ EVITANDO CHOCAR CON LAS"
54 PRINT"MINAS OCULTAS O CON EL MURO."
56 PRINT"■EN LA PARTE SUPERIOR DE LA PANTALLA SE"
58 PRINT"LE INDICA LAS MINAS QUE HAY ALREDEDOR"
60 PRINT"DE LA FIGURA (♠)."
62 PRINT"■■LOS CORAZONES (♥) QUE APARECEN A PARTIR"
64 PRINT"DE LA 3 PARTIDA LE DARA PUNTOS SI CON-"
66 PRINT"SIGUE TOCARLOS."
68 PRINT"■PARA MOVER LA FIGURA UTILICE LAS TECLAS"
```

Concurso

Viene de la página anterior

```
70 PRINT"  Z-DERECHA             M-ARRIBA"
72 PRINT"  X-IZQUIERDA           N-ABAJO"
74 PRINT"██████PULSE UNA TECLA"
76 GETT$:IFT$=""THEN76
77 PRINT"█"
78 PRINT"HAY DOS ZONAS SEGURAS,QUE SON:"
80 PRINT"DEBAJO DEL MURO SUPERIOR Y"
82 PRINT"ENCIMA DEL MURO INFERIOR."
84 PRINT"██DESPUES DE CADA JUEGO PODRA VER LA"
86 PRINT"LA REPETICION DE LA JUGADA"
88 PRINT"██EL MAXIMO DE JUEGOS ES DE 8"
90 PRINTTAB(13)"███BUENA SUERTE"
92 PRINT"█████PULSE CUALQUIER TECLA PARA COMENZAR"
94 GETR$:IFR$=""THEN94
96 PRINT"█"
180 PRINT"█"
190 GOSUB200:GOTO410
200 FORP=1TO38
210 P1=1024+P+40*3
220 POKEP1,102:POKEP1+760,102
230 NEXT
240 FORP=3TO22
250 P2=1024+1+40*P
260 POKEP2,102:POKEP2+37,102
270 NEXT
280 FORP=1163TO1165
290 POKEP,158:POKEP+760,32
300 NEXT
305 XR=28:YR=2:Z$="PARTIDAS-":GOSUB330:POKE1101,CO+49
310 XR=1:YR=2:Z$="█MINAS:-":GOSUB330
330 POKE214,YR:POKE211,XR-1:PRINTCHR$(145);Z$:RETURN
410 N=40:SC=0:CO=0:PA=0
412 XR=3:YR=5:Z$="██-------------ZONA SEGURA-------------":GOSUB330
414 XR=3:YR=22:Z$="-------------ZONA SEGURA-------------":GOSUB330
420 XR=13:YR=24:Z$="█PONIENDO MINAS":GOSUB330
430 T=0:T1=0
440 FORV=0TON
450 X(V)=INT(RND(1)*36)+2
460 Y(V)=INT(RND(1)*16)+5
470 M1=1024+X(V)+40*Y(V)
480 M2=55296+X(V)+40*Y(V)
490 POKEM2,1:POKEM1,81
500 NEXTV
510 IFCO>1THENZ1=INT(RND(1)*12)+7:Z2=1038+40*Z1:Z3=55310+40*Z1
515 IFCO>1THEN POKEZ2,83:POKEZ3,2:POKEZ2+11,83:POKEZ3+11,2
520 XR=12:YR=24:Z$="                ":GOSUB330
522 XR=3:YR=5:Z$="                                       ":GOSUB330
524 XR=3:YR=22:Z$="                                       ":GOSUB330
530 POKE1924,65
540 X=20:Y=23
550 FORS=1TO1000
560 R1=0:R2=0:R3=0:R4=0
570 GETA$:IFA$=""THEN570
580 IFA$="Z"THENX=X-1
```

```
590 IFA$="X"THENX=X+1
600 IFA$="M"THENY=Y-1
610 IFA$="N"THENY=Y+1
620 T=T+1
630 L(S)=1024+X+40*Y
635 IFL(S)>1943THENL(S)=1924
640 CH=PEEK(L(S))
650 IFCH=81ORCH=102THEN800
660 IFCH=158THEN1000
670 IFCH=83THENSC=SC+500
680 POKEL(S),65:POKEL(S-1),160
685 IFCO=6ANDT>45THENPOKEL(S-1),32
687 IFCO=6ANDT=45THENXR=7:YR=24:Z$="▓YA NO DEJAS HUELLAS":GOSUB330
690 IFPEEK(L(S)-1)=81THENR1=1
700 IFPEEK(L(S)+1)=81THENR2=1
710 IFPEEK(L(S)-40)=81THENR3=1
720 IFPEEK(L(S)+40)=81THENR4=1
730 POKE1071,176+R1+R2+R3+R4
740 NEXT
800 YR=12:XR=10:Z$="▓▓HAS SIDO DESTRUIDO":GOSUB330
810 SC=SC+100-T
820 FORES=0TO3000:NEXT:GOSUB830:GOTO950
830 PRINT"▓"
840 GOSUB200
850 FORV=0TON
860 M1=1024+X(V)+40*Y(V):M2=55296+X(V)+40*Y(V)
870 POKEM1,81:POKEM2,0
880 NEXT
885 IFCO>1THENPOKEZ3,5:POKEZ3+11,5:POKEZ2,83:POKEZ2+11,83
890 FORS=1TO1000
900 POKEL(S),65:POKEL(S-1),160
910 T1=T1+1
920 IFT1=T+1THENRETURN
930 FORES=0TO500:NEXT
940 NEXT
950 PRINT"▓"
955 PRINT"▓▓LO SIENTO NO HAS SUPERADO LAS 8 PRUEBAS"
960 PRINT"▓TU PUNTUACION ES DE";SC;"PUNTOS"
970 GOTO1300
1000 YR=12:XR=10:Z$="▓▓LO HAS CONSEGUIDO":GOSUB330
1020 SC=SC+300-T
1030 FORES=0TO3000:NEXT:GOSUB830
1040 YR=14:XR=6:Z$="▓TE ATREVES CON OTRA?-S(SI)/K(NO)":GOSUB330
1050 GETT$:IFT$=""THEN1050
1060 IFT$="S"THEN1080
1070 IFT$="K"THEN950
1075 GOTO1050
1080 PRINT"▓"
1090 CO=CO+1:POKE53280,2+CO
1100 N=N+10:PA=PA+1:IFPA>7THEN1200
1110 GOSUB200:GOTO420
1200 PRINT"▓"
1210 PRINT"▓MAGNIFICO,HAS SUPERADO LAS 8 PRUEBAS"
1220 PRINT"TU PUNTUACION FINAL ES:";SC;"PUNTOS"
```

Concurso

Viene de la página anterior




```
1230 GOTO1300
1300 PRINT"QQQQQOTRA PARTIDA SI/NO?"
1310 INPUTB$
1315 IFB$="SI"THENV=0:X=0:Y=0:L=0:CO=0:GOTO180
1320 IFB$="NO"THENEND
1325 PRINT"HABLA CLARO !!CARAMBA!!
1330 GOTO1300
```

Mensaje castellano

CBM 64

No todo van a ser juegos en el campo de los programas, por ello presentamos aquí este programita de utilidad o podríamos decir de "curiosidad" que recibimos de Clemente Pérez desde Zaragoza. El programa se encarga de traducir al castellano los famosos mensajes de error, del tipo Syntax error, que suelen aparecer cada dos por tres para fastidiarnos. El programa, que incluye alguna rutina en lenguaje máquina, se borra nada más ejecutado, apareciendo en la pantalla el mensaje LISTO en lugar de *READY*.

Después ya podemos equivocarnos sin problemas, ya que en lugar de fastidiarnos en inglés, el ordendor se encargará de fastidiarnos en castellano, lo cual es muy diferente. Este funcionamiento en castellano continuará indefinidamente hasta que apaguemos el ordenador o hasta que pulsemos las teclas *RUN/STOP* restore, con lo que volverán a aparecer las frases en inglés. En este segundo caso, es decir, al emplear *RUN/STOP RESTORE*, se puede volver al funcionamiento en castellano escribiendo *POKE* 1,53 y pulsando después RETURN.

```
  1 REM***********************************
  2 REM**MENSAJES DE ERROR EN CASTELLANO
  3 REM***********************************
  4 REM****   CLEMENTE PEREZ RODRIGUEZ
  5 REM***********************************
  7 REM########################
  8 REM## SUB L/M COPIA ROM
  9 REM########################
 10 FORI=0TO29:READJ:POKE828+I,J:NEXT
 15 SYS828
 20 POKE829,0:POKE834,224:POKE837,224
 25 SYS828
 40 DATA160,192,162,0,189,0,160,157,0,160,224,255,240,4,232,76,64,3,238,66,3
 42 DATA238,69,3,204,66,3,208,229,96
 45 REM########################
 46 REM## DATA MENSAJES ERROR
 47 REM########################
 50 READA:IFA=0THEN200
 52 IFA<256THENPOKEB,A:GOTO50
 55 READA$
 65 B=A
 70 FORN=1TOLEN(A$)
 80 POKEB,ASC(MID$(A$,N,1))
 85 B=B+1
 90 NEXTN
 95 GOTO50
 100 DATA61630,E/S,61642,BUSCANDO ,61652, A ,61657
 110 DATAPULSE TECLA PLAY ,160,61675,PULSE TECLAS RECORD Y PLAY,160
```

```
120 DATA61703,CARGO ,160,61711,GRABO ,61719,VERIFICO,160,61729,HALLO,61736,YA
130 DATA44285,IGNORO EXTRAS,44301,VOLVER A INICIO
140 DATA41842,E,41848,LISTO,41859,PARO ,41374,MUCHOS FICHERO,211
145 DATA41389,FICHERO ABIERT,207,41404,FICHERO CERRAD,207
150 DATA41419,FICHERO NO HALLAD,207,41437,FALTA PERIFERIC,207
155 DATA41453,FICHERO SALID,193,41467,FICHERO ENTRAD,193
160 DATA41482,FICHERO OMITID,207,41497,NUM. PERIF. ILEGA,204
165 DATA41515,NEXT SIN FO,210,41527,SINTAXIS MA,204,41539,RETURN SIN GOSU,194
170 DATA41555,NO HAY DAT,193,41566,CANTIDAD ILEGA,204,41581,NO CAB,197
175 DATA41588,MEMORIA LLEN,193,41601,LINEA INDEFINID,193,41617,INDICE FALS,207
180 DATA41629,MATRIZ REDI,205,41641,DENOMINADOR NUL,207,41657,DIRECTO ILEGA,204
185 DATA41671,CARACTER FALS,207,41685,CADENA LARG,193
190 DATA41697,DATA MAL,207,41706,FORMULA COMPLEJ,193,41722,NO PUEDO CON,212
195 DATA41735,FUNCION INDE,198,41748,NO COINCID,197,41759,CARGA MA,204,0
197 REM########################
198 REM### CAMBIO VECTORES
199 REM########################
200 FORX=41770TO41824STEP2
210 READY:W=PEEK(X)+Y
212 IFW>255THENW=W-256:POKEX+1,PEEK(X+1)+1
215 IFW<0THENW=W+256:POKEX+1,PEEK(X+1)-1
217 POKEX,W
220 NEXTX
250 DATA1,7,9,13,11,11,11,9,6,2,8,4,4,3,2,2,1,0
255 DATA-1,-1,-1,0,-3,-3,-6,-7,-10,-5
279 REM###############
280 POKE1,53:NEW
281 REM###############
```

PREMIADO CON 5.000 PESETAS

Mastermind

CBM-64

Juan Luis Aguado de Valladolid nos envía este juego para el C-64, MASTERMIND, una versión bastante clásica del conocido juego del mastermind. Se trata de adivinar una combinación de seis colores pensada en secreto por el ordendor. Se pueden realizar únicamente seis intentos para acertar con la combinación, teniendo en cuenta que ninguno de los colores puede estar repetido.

Los heridos y los muertos se representan por círculos blancos y negros respectivamente en la parte derecha de la pantalla, al lado de cada jugada. Como se sabe, los heridos representan un color acertado, pero que no está colocado en su lugar correcto, mientras que los muertos representan aciertos completos, es decir, tanto en el color como en la posición. El juego termina o cuando se consiguen seis muertos, es decir, que se ha acertado la combinación, o bien cuando se acaban los seis intentos.

```
1 REM
2 REM ****** DECLARACION DE VARIABLES**
3 REM ******     Y DIMENSIONES     **
4 REM
10 DIM P(6):DIM V(6):DIM C(4)
20 A=55540:C=1269:D=1886:T=VAL(RIGHT$(TI$,6))
24 REM
25 REM ****** PRESENTACION ************
26 REM
30 PRINT"♥":POKE 53281,12:POKE 53280,11
40 PRINT"♥▮▮▮▮▮▮▮MASTER  MIND":PRINT"▮▮▮▮▮┌─────────────"
50 PRINT"▮▮▮▮◣┌────────┬────┐      ┌──────────┐"
```

Concurso

Viene de la página anterior

```
60 PRINT"▮▮▣1-◣|▥ O O O◣|▥OOOO◣|     |            |"
70 PRINT"▮▮▮▮|├─────────┼─────┤     |            |"
80 PRINT"▮▮▣2-◣|▥ O O O◣|▥OOOO◣|     |■        ◣|"
90 PRINT"▮▮▮▮|├─────────┼─────┤     └────────────┘"
100 PRINT"▮▮▣3-◣|▥ O O O◣|▥OOOO◣|"
110 PRINT"▮▮▮▮|├─────────┼─────┤        ▣K   CRSOR   >"
120 PRINT"▮▮4-◣|▥ O O O◣|▥OOOO◣|"
130 PRINT"▮▮▮▮|├─────────┼─────┤"
140 PRINT"▮▮▣5-◣|▥ O O O◣|▥OOOO◣|     ▮PARA SELECCIONAR"
150 PRINT"▮▮▮◣├─────────┼─────┤"
160 PRINT"▮▮▣6-◣|▥ O O O◣|▥OOOO◣|     ▮EL COLOR PULSA"
170 PRINT"▮▮▮◣└─────────┴─────┘
180 PRINT"                              ▣▮ESPACE▮"
190 FOR N=0TO10:FORM=0TO 1:POKEA+20+N+40*M,15:POKE C+19+N+40*M,160:NEXTM:NEXTN
200 POKE A+20,7:POKE A+22,8:POKE A+24,10:POKEA+26,2:POKEA+28,5:POKE A+30,6
230 REM
240 REM ****** COMBINACION DEL ********
250 REM ******    ORDENADOR    ********
260 REM
300 FOR N=1 TO 4:C(N)=INT (RND(T)*7+2)
310 IF C(N)=3 THEN C(N)=8
320 IF C(N)=4 THEN C(N)=10
330 NEXT
340 IFC(2)=C(1)ORC(3)=C(2)ORC(3)=C(1)ORC(4)=C(3)ORC(4)=C(2)ORC(4)=C(1)THENGOTO30
0
400 REM
410 REM ****** TOMA DE DATOS **********
420 REM
500 POKE C+99+H,30:POKE C+99+(H-2),32:POKEC+99+(H+2),32
520 GET A$:IF A$="▮" THEN H=H+2
540 IF A$="▣" THEN H=H-2
560 IF A$=" " THEN GOSUB 2000
580 IF H>10 THEN H=10
600 IF H<0 THEN H=0
650 GOTO 500
2000 P(X)=PEEK (A+20+H) AND 15
2010 FOR B=0 TO 6 STEP 2:V(B)=PEEK(55541+B+40*Y)AND 15:NEXT B
2020 IF V(0)=P(X) OR V(2)=P(X) OR V(4)=P(X) OR V(6)=P(X) THEN GOTO 500
2030 POKE C+X+40*Y,81:POKE A+1+X+40*Y,P(X)
2040 X=X+2:IFX<=6 THEN:RETURN
2100 REM
2110 REM ****** COMPARACIONES *********
2120 REM
2200 IF P(0)=C(1) THEN BK=BK+1
2210 IF P(2)=C(2) THEN BK=BK+1
2220 IF P(4)=C(3) THEN BK=BK+1
2230 IF P(6)=C(4) THEN BK=BK+1
2240 IF P(0)=C(2) OR P(0)=C(3) OR P(0)=C(4) THEN WT=WT+1
2250 IF P(2)=C(1) OR P(2)=C(3) OR P(2)=C(4) THEN WT=WT+1
2260 IF P(4)=C(1) OR P(4)=C(2) OR P(4)=C(4) THEN WT=WT+1
2270 IF P(6)=C(1) OR P(6)=C(2) OR P(6)=C(3) THEN WT=WT+1
2300 REM
2310 REM ****** RESPUESTA DEL *********
2320 REM ******    ORDENADOR  *********
2340 REM
2500 FOR I=1 TO BK:POKE C+7+I+40*Y,81:POKE A+8+I+40*Y,0:NEXT I
2510 IF BK=4 THEN GOTO 4000
```

```
2520 IF WT=0 THEN GOTO 2540
2530 FOR J=1 TO WT:POKE C+7+BK+J+40*Y,81:POKE A+8+BK+J+40*Y,1:NEXT J
2540 Y=Y+2:IF Y<=10 THEN X=0:BK=0:WT=0:GOTO 500
2600 REM
2610 REM ****** MENSAJES FINALES ******
2620 REM
3000 PRINT"▮■■LO SIENTO,NO LO LOGRASTE."
3010 PRINT"■■■LA COMBINACION ERA:"
3020 FOR N=1 TO 4:POKE D+P,81:POKE56158+P,C(N):P=P+2:NEXT N
3030 GOTO 4010
4000 PRINT"▮■■ENHORABUENA,ACERTASTE."
4010 PRINT"■■■OTRA PARTIDA? (S/N)"
4020 GET A$:IF A$="" THEN 4020
4030 IF A$="S" THEN RUN
4040 PRINT"■■■OK. HASTA OTRA"
```

PROGRAMA commodore Magazine COMPROBADO

Simon

Esta versión del popular Simón para el *VIC-20* ha sido creada por Antonio Font, lector de Commodore Magazine de Barcelona. Simón es un juego que pone a prueba la retentiva y los reflejos del jugador, ya que consiste en recordar series cada vez mayores de colores y sonidos, para repetirlas a continuación, cada una más deprisa que la anterior. El programa que nos remite Antonio permite jugar de dos formas diferentes; un solo jugador contra el *VIC-20* o entre varios jugadores.

En la primera modalidad el *VIC-20* va mostrando al jugador, sucesivas series de colores. Cada serie repite todos los colores de la anterior y añade uno nuevo, al mismo tiempo que la velocidad aumenta. Después de cada serie el jugador debe intentar repetirla sin equivocarse con ninguno de los colores y dentro de un tiempo límite, para lo cual utilizará las teclas 1, 2, 3 y 4 que corresponden a los cuatro colores del juego. Vencer a la máquina es una tarea harto difícil, ya que es capaz de presentarle al jugador series de hasta **250 colores.**

Cuando se juega entre varios jugadores, lo primero es decidir cuántos van a ser, diciéndoselo al ordenador. A continuación y por turno cada uno de los jugadores irá repitiendo la serie de colores del jugador anterior añadiendo, al final, un color más. Cuando el último jugador haya terminado con su serie de colores, es el turno del primer jugador, que deberá hacer lo mismo que sus compañeros, con lo que se cierra el ciclo. El juego discurre de esta manera hasta que alguno de los jugadores se equivoca.

Al principio es muy sencillo recordar las series mientras son cortas, pero cuando se hacen mayores, el juego puede llegar a desencadenar auténticos ataques de histeria y si no, lo mejor es que cada uno lo compruebe por sí mismo.

VIC - 20

```
0 REM  **************
1 REM *             *
2 REM *   HECHO     *
3 REM *       POR   *
4 REM *             *
5 REM * ANTONIO     *
6 REM *        FONT *
7 REM *             *
8 REM *  BARCELONA  *
9 REM *             *
10 REM * [28-5-84]  *
11 REM *            *
12 REM **************
13 REM
```

Concurso

Viene de la página anterior

```
14 PRINT"■":S1=36874:S2=36876:S3=36877:S4=36875:C1=36879
15 GOSUB17:V=0:B=0:POKE36878,15
16 GOTO18
17 PRINT"■■■          SIMON          ■":POKEC1,88:RETURN
18 PRINT"■■■■ELIGE ENTRE ESTAS    ■■■■TRES OPCIONES"
19 PRINT"■■■1■ JUGAR CONTRA LA    ■■■■MAQUINA ■■(1 PERSONA)■"
20 PRINT"■■■2■ JUGAR CONTRA          ■■■■AMIGOS"
21 GETS$:IFS$="1"THENGOSUB90:GOTO24
22 IFS$="2"THENGOSUB90:GOTO28
23 GOTO21
24 GOSUB17:PRINT"■■■■CONSISTE EN ACERTAR     ■■■LOS MAS POSIBLES        ■■■COLORES
 ."
25 PRINT"■■■INTENTA BATIR EL        ■RECORD."
26 PRINT"■■■Y SI LLEGA AL FINAL.  ■■■!!!PREMIO!!!"
27 AA=0:FORX=1TO10000:NEXT:GOTO40
28 GOSUB17:PRINT"■■■■CONSISTE EN QUE EL     ■PRIMERO PULSE UN       ■COLOR CADA VEZ
 MAS"
29 PRINT"■■■Y QUE EL OTRO LOS       ■ACIERTE"
30 PRINT"■■Y SUME UNO MAS Y ASI ■SUCESIVAMENTE          ■    !!!SUERTE!!!"
31 INPUT"■JUGADORES";BB
32 A$="":PRINT"■":AA=1:C=1:E=1:IFB=0THEN37
33 C=1:IFAA=BB+1THENAA=1
34 FORX=1TOB:GOSUB52
35 I=0:PRINT"■":GOSUB80:GOSUB60
36 IFB$=MID$(A$,C,E)THENC=C+1:NEXT
37 GOSUB52:GOSUB80:AA=AA+1
38 A$=A$+B$:B=LEN(A$):IFB=250THENGOSUB63:GOTO69
39 GOTO33
40 A$=""
41 PRINT"■":A=INT(RND(1)*4)+1:E=1
42 IFA=1THENA$=A$+"1"
43 IFA=2THENA$=A$+"2"
44 IFA=3THENA$=A$+"3"
45 IFA=4THENA$=A$+"4"
46 B=LEN(A$)
47 IFB=250THENGOSUB63:GOTO69
48 FORC=1TOB
49 B$=MID$(A$,C,E)
50 GOSUB80:NEXT
51 FORC=1TOB:GOSUB52:GOTO57
52 B$="":J=1200-(B*4)
53 GETB$:I=I+1:PRINT"■■■■TIEMPO PARA LLEGAR A ■";J;"■■■■■■■■■■■";I:IFI>=JTHEN66
54 IFAA<>0THENPRINT"■■■■■■■■■■■■■CONCURSANTE ";AA
55 IFB$="1"ORB$="2"ORB$="3"ORB$="4"THENRETURN
56 GOTO53
57 PRINT"■":GOSUB80:GOSUB60
58 IFB$=MID$(A$,C,E)THENIFC=BTHENFORH=1TO500:NEXT:I=1:GOTO41
59 NEXTC
60 IFB$<>MID$(A$,C,E)ANDB>=50THENGOSUB63:V=1:GOTO71
61 IFB$<>MID$(A$,C,E)THENGOSUB63:GOTO71
62 RETURN
63 G=(B*5)-5:IFY<GTHENY=G
64 IFAA<>0THENPRINT"■■■■■■■■■■■■■■■■CONCURSANTE";AA
65 PRINT"■■■■■■■■■■■■CONSIGUISTE:";G;"P.":PRINT"■RECORD       :";Y;"P.":RETURN
```

```
66 PRINT"■■■■■MAS RAPIDO!!!"
67 PRINT"■LLEGASTES HASTA MI     ■LIMITE DE PACIENCIA   ■LO SIENTO"
68 GOSUB63:GOSUB86:GOTO75
69 PRINT"■■■■■ESTUPENDO!!!"
70 PRINT"■LLEGASTES A GASTAR MI ■MEMORIA. ME HE PERDI- ■DO YO":GOSUB88:GOTO75
71 PRINT"■■■■■FALLASTE!!!"
72 PRINT"■LO SIENTO NO ERA ESTE ■COLOR,ERA EL ";MID$(A$,C,E)
73 IFV=1THENGOSUB88:GOTO75
74 GOSUB86:GOTO75
75 PRINT"■■■■■■■■■■■■■■■■■■VOLVER A EMPEZAR [S,N]"
76 GETD$
77 IFD$="S"THENGOSUB90:GOTO14
78 IFD$="N"THENPRINT"■":POKEC1,27:GOSUB90:END
79 GOTO76
80 PRINT"■":POKES4,215:IFB$="1"THENPOKEC1,8:POKES2,240:PRINT"■";
81 IFB$="2"THENPOKEC1,25:POKES2,151:PRINT"■";
82 IFB$="3"THENPOKEC1,42:POKES2,201:PRINT"■";
83 IFB$="4"THENPOKEC1,59:POKES2,227:PRINT"■";
84 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■";B$
85 FORH=1TO800-(B*2.75):NEXT:POKEC1,88:POKES2,0:PRINT"■■":POKES4,0:RETURN
86 FORX=241TO135STEP-0.5:POKES3,X:NEXT:POKES3,0:POKES2,239
87 FORX=1TO100:NEXT:POKES2,0:RETURN
88 FORP=1TO10:FORX=229TO241:POKES1,X:NEXT:FORX=241TO229STEP-1:POKES1,X:NEXT:NEXT
89 POKES1,0:POKES2,239:FORX=1TO100:NEXT:POKES2,0:RETURN
90 POKES2,200:FORX=1TO300:NEXT:POKES2,0:RETURN
```

Gráficos con

Uno de los campos de mayor auge en la programación de ordenadores personales es, sin género de dudas, el desarrollo de juegos. Estos programas se caracterizan, por lo general, por un empleo a fondo de las capacidades gráficas y de sonido del ordenador. Además, un buen programa de juegos sólo se consigue con mucho trabajo, empleando todos los trucos de la máquina y conociendo a fondo todos los recovecos de la memoria, las rutinas del sistema operativo y un largo etcétera. De todas formas no hay que asustarse, ya que se pueden conseguir programas interesantes sin necesidad de llegar a estos extremos.

En este artículo vamos a ver cómo manejar los gráficos y dibujar en la pantalla mediante instrucciones PEEK y POKE, en lugar de utilizar instrucciones PRINT. Esta forma de actuar es uno de los truquitos empleados en la mayoría de los juegos, y además, el llegar a explicar su funcionamiento nos va a permitir hablar de cosas tan interesantes como la estructura de las memorias de pantalla y del color.

LA MEMORIA DE PANTALLA

La técnica de imprimir en la pantalla mediante instrucciones POKE se basa en que, todo los caracteres o gráficos que aparecen en la pantalla en un instante dado, están contenidos en una zona de memoria RAM denominada memoria de pantalla. Si nosotros mediante un POKE escribimos algo en la memoria de pantalla, entonces ese algo aparecerá en la pantalla del televisor. De la misma forma, al leer con PEEK una posición de la memoria de pantalla podemos saber qué carácter hay depositado en la posición correspondiente de la pantalla del televisor. Esta forma de actuar tiene la ventaja, sobre la instrucción PRINT, de ser mucho más rápida en la mayoría de los casos.

La decisión sobre si conviene emplear sentencias PRINT o escribir directamente en la memoria de pantalla, dependerá en general del tipo de programa. Sobre esto no hay reglas

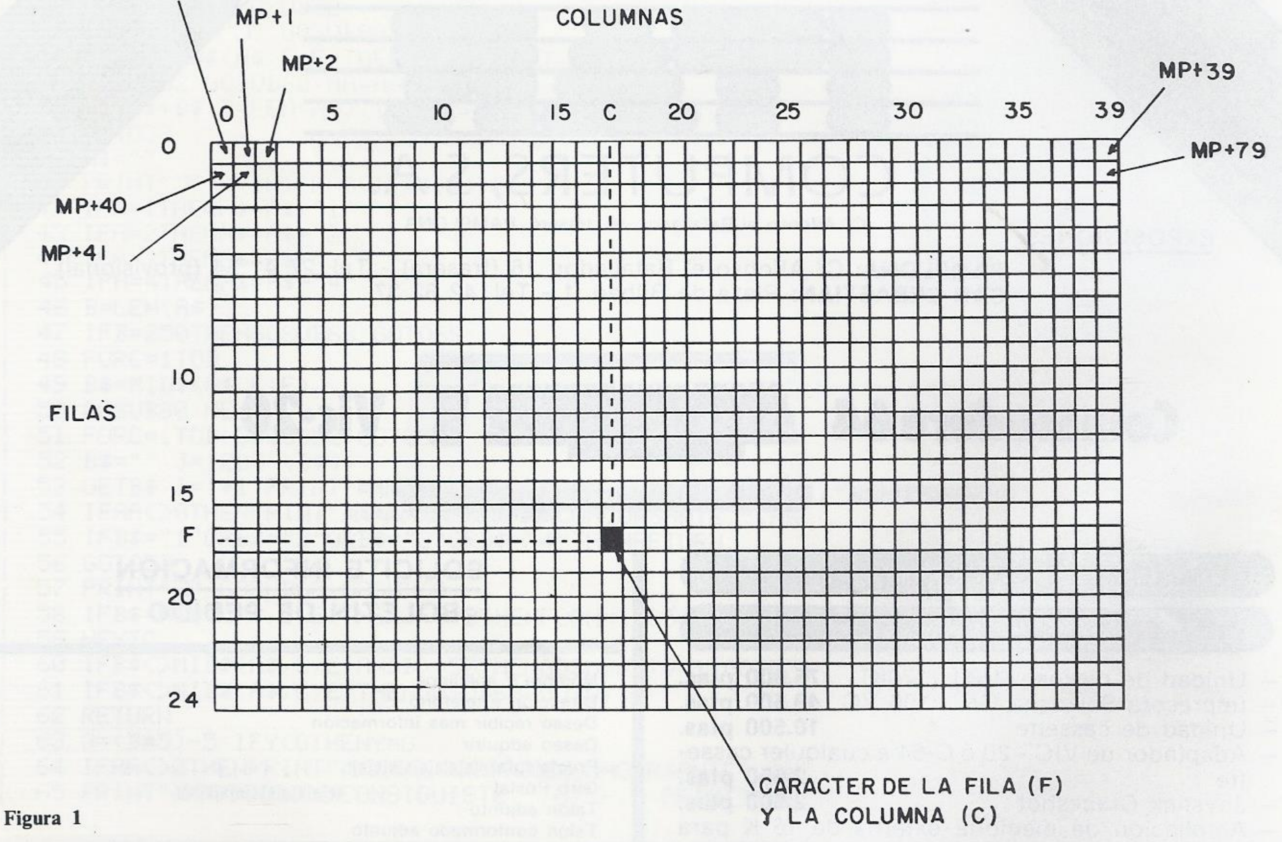


Figura 1

LA OTRA FORMA DE LEER EL MANUAL

Peek y Poke

fijas y normalmente será la experiencia del programador la que le permitirá escoger la forma más adecuada. Como norma, la instrucción PRINT es útil y rápida cuando se trata de imprimir grandes cantidades de datos o gráficos muy extensos. El problema surge cuando queremos imprimir caracteres o gráficos en posiciones determinadas de la pantalla, pues aunque la instrucción PRINT es bastante rápida a la hora de imprimir, es sin embargo lenta cuando se trata de desplazar el cursor por la pantalla y situarlo en una posición determinada, antes de llevar a cabo la impresión.

Este problema se ve agravado porque el BASIC del **Commodore 64** no dispone de la capacidad AT, disponible en otras versiones del BASIC, y que utilizada en una sentencia PRINT permite realizar la impresión en cualquier posición de la pantalla sin más que indicar las coordenadas X e Y (fila y columna) del punto donde va a comenzar la impresión. Esta lentitud es la principal razón por la que no se puede conseguir un juego de acción demadiado rápido escribiendo el programa en BASIC.

Sin embargo, utilizando instrucciones POKE es posible mover pequeñas cantidades de datos por la pantalla a una velocidad aceptable.

Para poder escribir directamente sobre la memoria de pantalla hay que saber cómo está organizada y qué direcciones de memoria ocupa. La pantalla del **C-64** consta de 25 filas y 40 columnas por lo que hay un total de 1.000 posiciones sobre las que se puede imprimir. El contenido de estas posiciones se guarda en 1.000 bytes de RAM, ordenados por filas, de forma que el byte de menor dirección se corresponde con el ángulo superior izquierdo de la pantalla, mientras que el de mayor dirección representa el ángulo inferior derecho de la misma. Esta organización tan sencilla puede verse en la figura 1. Cuando se enciende el ordenador, la memoria de pantalla comienza en la dirección 1024 ($0400 en hexadecimal), pero es posible, y a veces muy interesante, colocarla en cualquier otra zona de RAM.

Hay otra zona de memoria muy importante, con la misma estructura y organización que la memoria de pantalla aunque su dirección de comienzo es diferente, que se denomina memoria de color y en la que se almacena el color de cada una de las posiciones de la pantalla. Cuando queremos imprimir un carácter mediante una instrucción POKE, además de colocar el valor correspondiente al carácter en la memoria de pantalla, hay que introducir el color que se desee para el carácter en la memoria de color. Si esto no se lleva a cabo el carácter no podrá verse al tener el mismo color que el fondo de la pantalla. La memoria de color, que también está constituida por 1.000 bytes, ordenados por filas como ocurre en la memoria de pantalla, comienza en la dirección 55296 ($D800 en hexadecimal) y al contrario que la memoria de pantalla, no puede moverse de su lugar, es decir, no es reubicable.

Vamos a ver cómo calcular las direcciones de la memoria de pantalla y de la memoria de color sobre las que hay que escribir. Para empezar vamos a identificar cada posición de la pantalla por su fila (F) y su columna (C), tal y como hemos visto en la figura 1. Entonces, y sin más que observar dicha figura, es muy sencillo comprobar que las direcciones de la

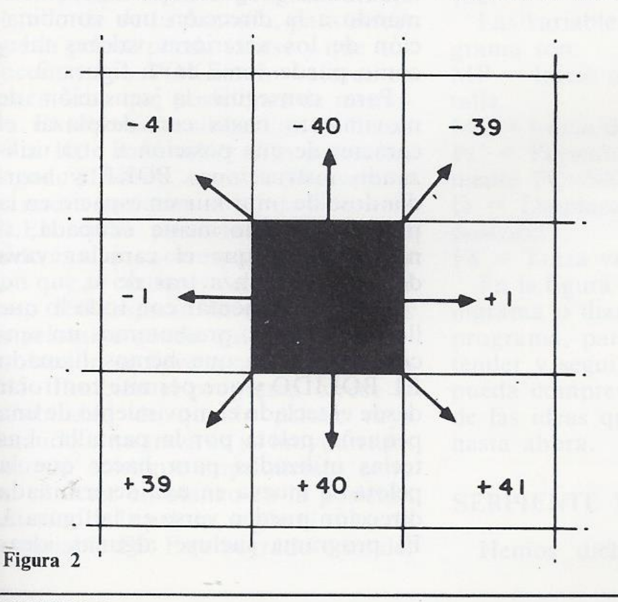


Figura 2

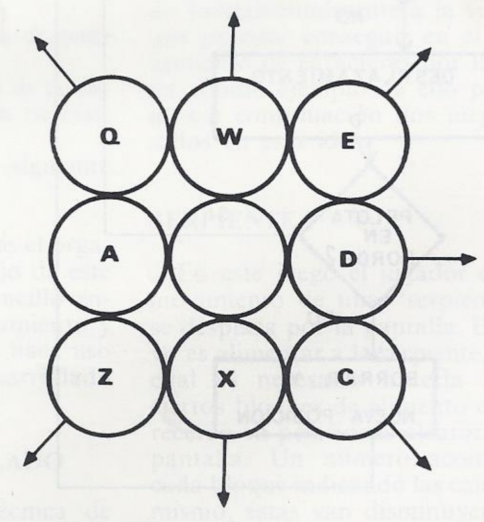


Figura 3

memoria de pantalla y de la memoria de color correspondientes a la posición de pantalla de la fila F y la columna C se obtienen de la siguiente forma:
Dirección en la memoria de pantalla:
DIRECCION=INICIO MEMORIA DE PANTALLA + C + 40*F
Dirección en la memoria de color:
DIRECCION=INICIO MEMORIA DE COLOR + C + 40*F

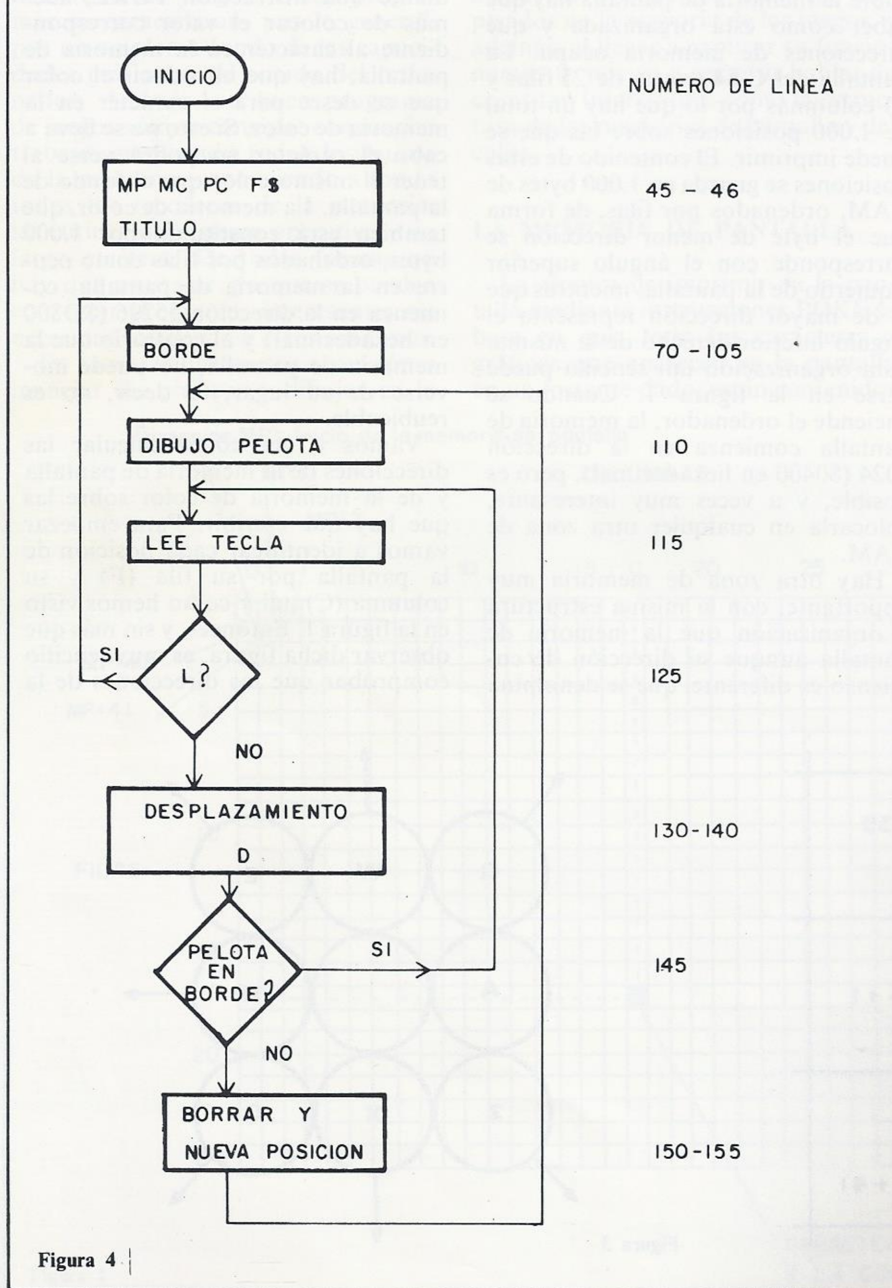


Figura 4

Siendo las direcciones de inicio de la memoria de pantalla y de la memoria de color las que ya hemos indicado.

Un aspecto importante que conviene señalar es que los códigos de carácter e introducir en la memoria de pantalla mediante POKE no son los códigos ASCII de la instrucción CHR$, sino que son valores que representan la posición del carácter a imprimir, dentro de la ROM generadora de caracteres. El motivo de utilizar estos valores, en lugar de los valores del código ASCII, está en que, de esta forma, el sistema puede acceder mucho más rápidamente a la información sobre la forma del carácter, al dirigirse directamente a la ROM generadora de caracteres. En la tabla 1 pueden verse los códigos de PEEK y POKE correspondientes a todos los caracteres. A la hora de escribir cualquier programa es muy importante tener en cuenta la existencia de estos dos códigos de caracteres diferentes, y no confundir entre uno y otro.

Una vez calculada la posición de la memoria de pantalla sobre la que hay que escribir el dato, como acabamos de ver, es muy sencillo dar un paso más que nos va a permitir movernos por toda la pantalla sin necesidad de recalcular de nuevo las direcciones de memoria correspondientes a las nuevas posiciones de la pantalla. En efecto, debido a la organización de las memorias de pantalla y de color, es muy sencillo calcular la dirección de las casillas adyacentes a una dada. Como puede verse en la figura 2, sumando a la dirección —1 ó +1 nos podemos desplazar una posición a la izquierda o a la derecha respectivamente, mientras que si sumamos —4 ó +4 podemos desplazarnos hacia arriba o hacia abajo. Los desplazamientos en diagonal se obtienen sumando a la dirección una combinación de los anteriores valores tal y como puede verse en la figura 2.

Para conseguir la sensación de movimiento basta con desplazar el carácter de una posición a otra utilizando instrucciones POKE y acordándose de imprimir un espacio en la posición anteriormente ocupada, si no se desea que el carácter vaya dejando una traza tras de sí.

Para experimentar con todo lo que llevamos dicho, presentamos un sencillo programa que hemos llamado EL BOLIDO y que permite controlar desde el teclado el movimiento de una pequeña pelota por la pantalla. Las teclas utilizadas para hacer que la pelota se mueva en una determinada dirección pueden verse en la figura 3. El programa incluye algunas ideas

```
10 REM ************
15 REM *          *
20 REM * EL BOLIDO *
25 REM *          *
30 REM ************
35 REM
40 REM
45 MP=1024:MC=55296:PC=500
50 PRINT"J"
55 PRINT"®L BORRA LA PANTALLA"
60 INPUT"QQQQTRAZA OCULTA (S/N) ";F$
65 PRINT"J"
70 FOR J=0 TO 39
75 POKE MP+J,160:POKE MP+J+960,160
80 POKE MC+J,0:POKEMC+J+960,0
85 NEXTJ
90 FOR J=0 TO 24
95 POKE MP+40*J,160:POKEMP+40*J-1,160
100 POKE MC+40*J,0:POKEMC+40*J-1,0
105 NEXT J
110 POKE PC+MP,81:POKEPC+MC,5
115 GET A$
120 IF A$=""THEN 145
125 IFA$="L"THEN65
130 D=(39ANDA$="Z")+(40 AND A$="X")+(41 AND A$="C"): IFD<>0THEN145
135 D=-(39AND A$="E")-(40 AND A$="W")-(41 AND A$="Q"):IF D<>0 THEN 145
140 D=(1 AND A$="D")-(1 AND A$="A")
145 IF PEEK(MP+PC+D)=160 THEN 115
150 POKE PC+MP,32+(49 AND F$="N")
155 PC=PC+D: GOTO 110
```

interesantes que vamos a comentar.

En primer lugar el progrma dibuja el reborde de la pantalla, para evitar que la pelota pueda salirse y que nos encontremos POKEando valores en quién sabe qué posiciones de memoria. El reborde se sibuja con las líneas 70 a 105. Para saber si la pelota ha llegado al borde, antes de dibujarla en su nueva posición, se lee mediante un PEEK dicha posición y si el valor leído es el que corresponde al carácter con que se construye el borde, en este caso 160 (espacio inverso), entonces la pelota se queda quieta. Esta comprobación se lleva a cabo en la línea 145. También el programa ofrece la posibilidad de que la pelota vaya dejando o no traza en su movimiento. Esto se consigue imprimiendo o no un carácter en blanco en la posición anterior de la pelota, según el valor de la variable F$. Este borrado o no del carácter se lleva a cabo en la línea 150.

Las variables utilizadas por el programa son:

MP = Inicio de la memoria de pantalla.
MC = Inicio de la memoria de color.
PC = Posición de la pelota (inicialmente PC=500)
D = Desplazamiento a la siguiente posición
F$ = Traza vista u oculta

En la figura 4 presentamos el organigrama o diagrama de flujo de este programa, para que sea sencillo entender y seguir su funcionamiento y pueda comprenderse cómo hace uso de las ideas que hemos desarrollado hasta ahora.

SERPIENTE Y ACORRALADO

Hemos dicho que la técnica de imprimir gráficos mediante instrucciones POKE se utiliza en un gran número de programas de juego, debido fundamentalmente a la velocidad que permite conseguir en el desplazamiento de caracteres por la pantalla. Como ejemplo de ello presentamos a continuación dos juegos, basados en esta idea.

SERPIENTE

En este juego el jugador dirige el movimiento de una "serpiente" que se desplaza por la pantalla. El objetivo es alimentar a la serpiente, para lo cual es necesario hacerla llegar a ciertos bloques de alimento que aparecerán en posiciones aleatorias de la pantalla. Un número acompaña a cada bloque indicando las calorías del mismo, éstas van disminuyendo con

el tiempo por lo que es importante que la serpiente se zampe cada bloque cuanto antes, antes de que pierda todo su valor alimenticio. A medida que se va alimentando, la serpiente va haciéndose más y más larga y con ello más difícil de controlar. El juego termina cuando la serpiente choca con el borde de la pantalla o con ella misma, apareciendo entonces la puntuación que ha conseguido el jugador.

Para controlar el movimiento se emplean únicamente dos teclas; "N" para girar a la izquierda y "M" para girar a la derecha.

ACORRALADO

Este segundo ejemplo del empleo de los gráficos con PEEK y POKE es un juego en el que hay que vencer al ordenador. Ambos, jugador y ordenador, van trazando una línea por la pantalla. El jugador controla el dibujo de la suya mediante las teclas:

W = Arriba
X = Abajo
A = Izquierda
D = Derecha

Hay que tener en cuenta que mientras no haya ninguna tecla pulsada, la línea continúa dibujándose en la

```
10 REM *************
15 REM *           *
20 REM * SERPIENTE *
25 REM *           *
30 REM *************
35 REM
40 DIM S(1000):A(1)=40:A(2)=1:A(3)=-40:A(4)=-1
45 SM=1024:CM=55296:HS=0
50 SC=0:T=1521:H=1525:L=5:J=1:PRINT"J"
55 FOR K=CM TO CM+999:POKEK,7:NEXTK
60 FOR K=SM TO SM+39:POKEK,160:POKEK+40,160:POKEK+960,160:NEXTK
65 FOR K=SM TO SM+960STEP40:POKEK,160:POKEK-1,160:NEXTK
70 FOR K=T TO H:POKEK,87:NEXTK
75 FOR K=1 TO L:S(K)=T+K-1:NEXTK
80 PH=L+1:PT=1
85 PRINT"■■■PUNTOS";SC;TAB(12);"LONGITUD";L;TAB(28);"RECORD";HS
90 GETK$
95 IF K$="N"THEN J=J+1:IFJ=5 THEN J=1
100 IF K$="M" THEN J=J-1:IFJ=0THEN J=4
105 H=H+A(J):SQ=PEEK(H):V=SQ-48
110 IF SQ=160 THEN 195
115 IF SQ=87 THEN 195
120 IF SQ=32 THEN 130
125 F=F+V:SC=SC+V:CV=0:L=L+V
130 S(PH)=H:PH=PH+1
135 POKEH,87
140 IF F<>0 THEN F=F-1 :GOTO 150
145 POKE S(PT),32:PT=PT+1
150 IF PT>1000 THEN PT=1
155 IF PH>1000 THEN PH=1
160 IF RND(0)<0.01 THEN CV=9
165 IFCV<10 THEN CV=0: IFPEEK(CP)<>87 THEN POKECP,32
170 IFCV<>0 THEN CV=CV-1:POKECP,INT(CV/10)+48:GOTO185
175 IF RND(0)<0.8 THEN 185
180 CV=100:CP=SM+1000*RND(0):IFPEEK(CP)<>32 THEN 180
185 FOR K=1 TO50:NEXTK:REM ESTE RETARDO PUEDE REDUCIRSE PARA MAYOR VELOCIDAD
190 GOTO 85
195 IF SC>HS THEN HS=SC
200 PRINT"■■■PUNTOS";SC;TAB(12);"LONGITUD";L;TAB(28);"RECORD";HS
205 FOR K=1 TO 200 :NEXTK
210 PRINT"■■■■■■■■■■■■PULSE ESPACIO PARA CONTINUAR"
215 GET K$:IFK$=" "THEN50
220 GOTO 215
```

```
10 REM *************
15 REM *           *
20 REM * ACORRALADO *
25 REM *           *
30 REM *************
35 REM
40 FOR J=1 TO 8:READA(J):NEXTJ
45 SM=1024:CM=55296:BT=0
50 DATA 40,-40,1,-1,40,-40,1,-1
55 PRINT"J":FORK=CM TO CM+999:POKEK,7:NEXTK
60 FOR K=SM TO SM+39:POKEK,160:POKEK+40,160:POKEK+960,160:NEXTK
65 FORK=SM TO SM+960 STEP 40:POKEK,160:POKEK-1,160:NEXTK
70 XM=A(INT(RND(0)*4+1)):YM=A(INT(RND(0)*4+1)):X=530:Y=545:T=TIME:M=1
75 POKESM+X,160:POKECM+X,3:POKESM+Y,160:POKECM+Y,5
80 FOR K=1 TO 70:REM *RETARDO PARA LA VELOCIDAD*
85 PRINT"▒▒▒▒▒▒▒▒▒▒▒TIEMPO";INT((TIME-T)/60);TAB(20);"MEJOR TIEMPO";BT
90 M=0:GETK$
95 IF K$="D" THEN M=1        "S"
100 IF K$="A" THEN M=-1      "A"
105 IF K$="W" THEN M=-40     "O"
110 IF K$="X" THEN M=40      "P"
115 IF M=-XM THEN 125
120 IF M<>0 THEN XM=M
125 X=X+XM
130 IF PEEK(SM+X)<>32 THEN GOTO 215
135 POKESM+X,160:POKECM+X,3
140 IF RND(0)>0.9 THEN 165
145 FOR J=1 TO 3
150 IF PEEK(SM+Y+J*YM)<>32 THEN 165
155 NEXTJ
160 GOTO 200
165 M=INT(RND(0)*4+1)
170 FOR P=1 TO 3
175 FOR Q=M TO M+3
180 YM=A(Q)
185 FOR R=1 TO 4-P
190 IF PEEK(SM+Y+R*YM)<>32 THEN 205
195 NEXT R
200 Y=Y+YM:POKESM+Y,160:POKECM+Y,5:GOTO85
205 NEXTQ:NEXTP
210 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒GANASTE!!":GOTO220
215 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒HE GANADO!!"
220 T=(TIME-T)/60:IF T>BT THEN BT=INT(T)
225 FOR T=1 TO 2000:NEXTT
235 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒PULSA ESPACIO PARA CONTINUAR "
240 GETK$:IFK$=" " THEN 55
245 GOTO 240
```

misma dirección que llevaba y que tampoco habrá ningún cambio de dirección si se intenta que la línea vuelva sobre sí misma, en dirección opuesta a la que hasta ese momento traía. El ordenador, por su parte, también va dibujando su propia línea. El objetivo del juego es sobrevivir tanto tiempo como sea posible sin tropezar con ninguna de las dos líneas ni con el reborde de la pantalla. Lo ideal es acorralar en un primer momento al ordenador y luego esperar tranquilamente, moviéndose con cuidado, a que se estrelle con nuestro cerco, pero claro, el ordenador no es tonto y acorralarle no es tan sencillo como pudiera parecer.

Con estos dos juegos hemos querido mostrar cómo, con la técnica de impresión de gráficos mediante PEEK y POKE, es posible conseguir efectos de movimiento muy interesantes y bastante rápidos, todo ello sin

emplear más que unas pocas instrucciones.

MODO DE EXTENSION DEL COLOR DE FONDO

Este modo es una forma un poco diferente de utilizar la memoria de pantalla y permite que el color de fondo de cada posición sea diferente, a elegir entre cuatro posibles colores. Para entrar en este modo hay que poner a "1" el bit 6 de la posición de memoria 53265 ($D011 en hexadecimal) que corresponde a un registro del *chip* VIC II (Controlador del *Interface* de Video). Desde un programa en BASIC la entrada y salida de este modo se pueden conseguir sencillamente mediante el siguiente par de instrucciones:

Para entrar en el modo:
POKE 53265,PEEK (53265) OR 64

Para salir del modo:
POKE 53265,PEEK (53265) AND 191

La desventaja de trabajar en este modo de extensión del color de fondo está en que sólo se pueden imprimir en la pantalla los primeros 64 caracteres de los reseñados en la tabla 1. Esto se debe a la forma de funcionamiento de este modo, ya que de los 8 bits de cada posición de la memoria de pantalla, sólo los 6 primeros se emplean para el código del carácter. Los dos bits que quedan, bits 6 y 7, se utilizan para seleccionar una entre cuatro posibilidades para el color de fondo del carácter.

¿Cuáles son los cuatro colores que se pueden emplear? Esto lo decide el programador, ya que puede escoger cuatro colores cualesquiera entre los 16 de que dispone el **C-64**. Lo único que tiene que hacer cuando haya seleccionado los cuatro colores es introducir en las posiciones de memoria 53281 a 53284 los números que

```
1 REM     CODIGOS DE PEEK Y POKE
2 REM     ***********************
3 REM
10 REM              @  0
11 REM              A  1
12 REM              B  2
13 REM              C  3
14 REM              D  4
15 REM              E  5
16 REM              F  6
17 REM              G  7
18 REM              H  8
19 REM              I  9
20 REM              J  10
21 REM              K  11
22 REM              L  12
23 REM              M  13
24 REM              N  14
25 REM              O  15
26 REM              P  16
27 REM              Q  17
28 REM              R  18
29 REM              S  19
30 REM              T  20
31 REM              U  21
32 REM              V  22
33 REM              W  23
34 REM              X  24
35 REM              Y  25
36 REM              Z  26
37 REM              [  27
38 REM              £  28
39 REM              ]  29
40 REM              ↑  30
41 REM              ←  31
42 REM       ESPACIO   32
43 REM              !  33
44 REM              "  34
45 REM              #  35
46 REM              $  36
47 REM              %  37
48 REM              &  38
49 REM              '  39
50 REM              (  40
51 REM              )  41
52 REM              *  42
53 REM              +  43
54 REM              ,  44
55 REM              -  45
56 REM              .  46
57 REM              /  47
58 REM              0  48
59 REM              1  49
60 REM              2  50
61 REM              3  51
62 REM              4  52
63 REM              5  53
64 REM              6  54
65 REM              7  55
66 REM              8  56
67 REM              9  57
68 REM              :  58
69 REM              ;  59
70 REM              <  60
71 REM              =  61
72 REM              >  62
73 REM              ?  63
74 PRINT"          ─  64
75 PRINT"          ♠  65
76 PRINT"          │  66
77 PRINT"          ─  67
78 PRINT"          ─  68
79 PRINT"          ─  69
80 PRINT"          ─  70
81 PRINT"          │  71
82 PRINT"          │  72
83 PRINT"          ╮  73
84 PRINT"          ╰  74
85 PRINT"          ╯  75
86 PRINT"          L  76
87 PRINT"          ╲  77
88 PRINT"          ╱  78
89 PRINT"          ┌  79
90 PRINT"          ┐  80
91 PRINT"          ●  81
92 PRINT"          _  82
93 PRINT"          ♥  83
94 PRINT"          │  84
95 PRINT"          ╭  85
96 PRINT"          ╳  86
97 PRINT"          ○  87
98 PRINT"          ♣  88
99 PRINT"          │  89
100 PRINT"         ♦  90
101 PRINT"         ┼  91
102 PRINT"         ▒  92
103 PRINT"         │  93
104 PRINT"         π  94
105 PRINT"         ◥  95
106 PRINT"ESPACIO     96
107 PRINT"         ▌  97
108 PRINT"         ▄  98
109 PRINT"         ▔  99
110 PRINT"         ▁  100
111 PRINT"         ▏  101
112 PRINT"         ▒  102
113 PRINT"         ▕  103
114 PRINT"         ▒  104
115 PRINT"         ◤  105
116 PRINT"         ▕  106
117 PRINT"         ├  107
118 PRINT"         ▗  108
119 PRINT"         └  109
120 PRINT"         ┐  110
121 PRINT"         ▂  111
122 PRINT"         ┌  112
123 PRINT"         ┴  113
124 PRINT"         ┬  114
125 PRINT"         ┤  115
126 PRINT"         ▎  116
127 PRINT"         ▍  117
128 PRINT"         ▕  118
129 PRINT"         ▔  119
130 PRINT"         ▀  120
131 PRINT"         ▃  121
132 PRINT"         ▟  122
133 PRINT"         ▖  123
134 PRINT"         ▝  124
135 PRINT"         ┘  125
136 PRINT"         ▘  126
137 PRINT"         ▚  127
```

representan a cada color, que son los indicados en la tabla 2. Supongamos que se decide trabajar con los cuatro primeros colores como colores de fondo. Entonces hay que introducir en las direcciones 53281 a 53284 y sin que sea excesivamente importante el orden, los valores 0, 1, 2, y 3 correspondientes a los colores elegidos. A partir de ahora donde hay que fijarse es en la tabla 3, donde se puede ver cómo obtener el código a POKEar en la memoria de pantalla. En esta tabla se ve por ejemplo, que para imprimir un carácter cuyo color de fondo sea el que hemos puesto en la posición 53283 (el rojo, por ejemplo, si en dicha posición hemos puesto el valor 2), el valor que hay que POKEar en la memoria de pantalla es el valor normal de la tabla 1. más 128 para poner "1" y "0" en los bits 7 y 6 en la memoria de pantalla. Todo esto, aunque parece complicado al principio es cuestión de que cada uno experimente por su cuenta y observe qué es lo que ocurre en cada caso, de esta forma y después de unos cuantos ensayos y algunas equivocaciones se comprenderá perfectamente el funcionamiento.

TABLA 2 (CODIGOS DE COLORES)
****

Color	Codigo	Color	Codigo
Negro	0	Naranja	8
Blanco	1	Marron	9
Rojo	2	Rosa	10
Cyan	3	Gris 1	11
Purpura	4	Gris 2	12
Verde	5	Verde claro	13
Azul	6	Azul claro	14
Amarillo	7	Gris 3	15

Bits 7 y 6 de la memoria de pantalla	Codigo de PEEK y POKE	Direccion del color de fondo
00	0+X	#0 53281 (D021)
01	64+X	#1 53282 (D022)
10	128+X	#3 53283 (D023)
11	192+X	#4 53284 (D024)

para todos LAPIZ OPTICO (y II)

El sencillo programa del capítulo anterior permite situar el carácter que seleccionemos por teclado en una posición definida de la pantalla, para ello se procede a rastrear la pantalla desde la posición 32900 hasta 32900 + 750 (con un asterisco). Si en estas condiciones situamos el lápiz sobre el mismo en cualquier instante de su recorrido, será automáticamente sustituido por el carácter seleccionado; el programa está preparado para funcionar en lazo continuo y se sugiere experimentar, modificándolo para aceptar caracteres especiales (sólo se han previsto los ASCII, habría que modificar las conversiones en la línea 1017) y extendiéndolo al total de la pantalla; haciendo que se visualize un mensaje alfanumérico de n caracteres, etcétera.

En cualquier caso, el estudioso comprobará que la principal limitación de este tipo de lápiz óptico controlado desde el BASIC es, lógicamente, la limitada velocidad y la muy pequeña inmunidad al ruido. Por lo que respecta a la velocidad, ésta puede ser incrementada en forma medianamente aceptable sin necesidad de acudir a la programación en código máquina, utilizando los compiladores disponibles DTL, PETSPEED e incluso, el simple y "misterioso" FASTER BASIC, que actúa sobre las comparaciones IF THEN y las GOTO, sin necesidad de compilar los programas.

Por desgracia, la eliminación del ruido (en este caso tanto el ruido real del integrado, como las señales indeseables producidas por reflejos, etc.) sólo puede ser alcanzada haciendo que el carácter que se utiliza como rastreador parpadee a elevada veloci-

dad, mientras se toma un número lo suficientemente elevado de lecturas del *Port* (5 por lo general), que permita por medias o mejor aún, por redundancia, eliminar las señales indeseables, en la fig. 7 se indica esquemáticamente el proceso:

LAPIZ OPTICO CONTROLADO POR HARDWARE

Los modelos de microordenadores más recientes cuentan con un *hardware* específico que puede determinar la posición real del SPOT sobre la pantalla, tal sucede con los modelos **Vic-20** y **64**, p. ejemplo, que utilizan para ello el sofisticado y complejo V.I.C. *(VIDEO INTERFACE CHIP)*, que controla y almacena, todos los datos del posicionamiento del lápiz óptico.

El *hardware* necesario para un lápiz de este tipo se diferencia fundamentalmente en no precisar de un efecto de LACHT (y por tanto, de una línea de Reset) para su correcto funcionamiento.

En la fig. 8 se presenta un sencillo circuito que permite realizar experiencias en este sentido:

Como puede verse en la misma se utiliza un Trigger de Schmitt que detecta el paso de T1 desde el estado de corte al de saturación por su entrada 10 (al emisor del transistor). Las entradas 13, 12 y 11 se encuentran fijadas a positivo mediante la resistencia R1 y el nivel de sensibilidad puede regularse con el P1 y aunque no alcanza valores elevados, resulta suficiente para la mayor parte de las aplicaciones.

Antes de proceder a la prueba de este circuito, es interesante comentar algunas características de las posiciones de memoria asociadas que se especifican en la fig. 9 (referidas al Vic-20).

A partir de esta información, una vez comprobado el funcionamiento del circuito del lápiz y su sensibilidad a la luz por métodos equivalentes a los descritos con anterioridad, se puede entrar un simple programa, como el indicado, que permite reali-

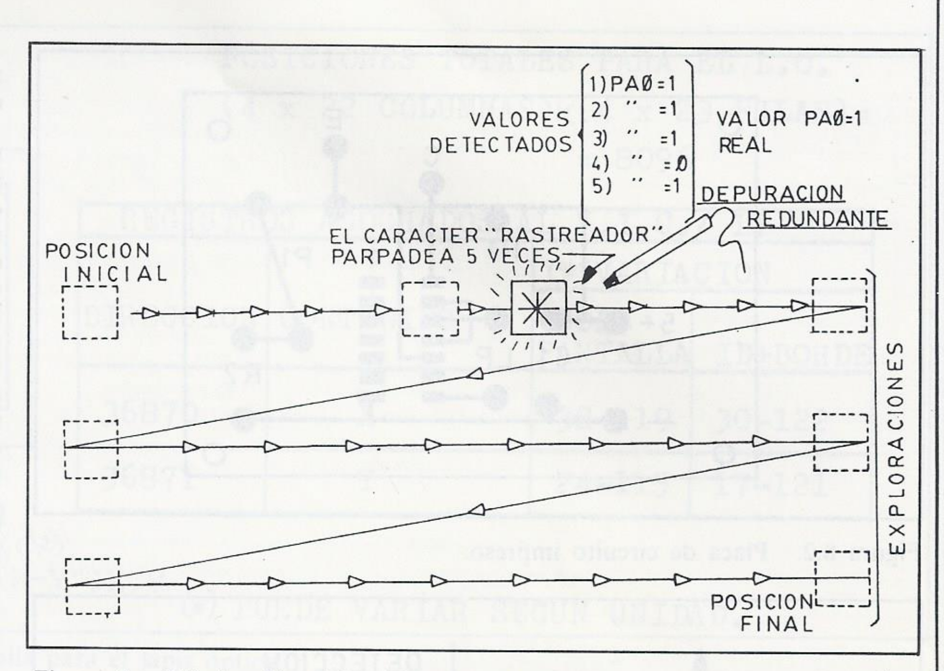


Figura 7. Eliminación del ruido por redundancia.

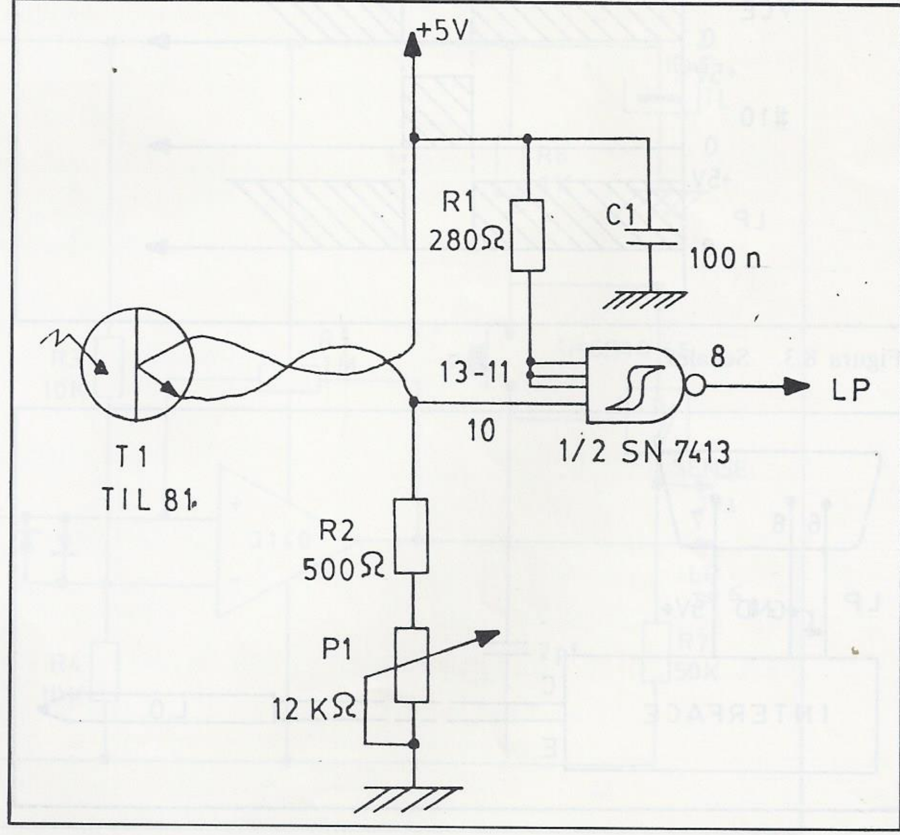


Figura 8.1. Circuitos.

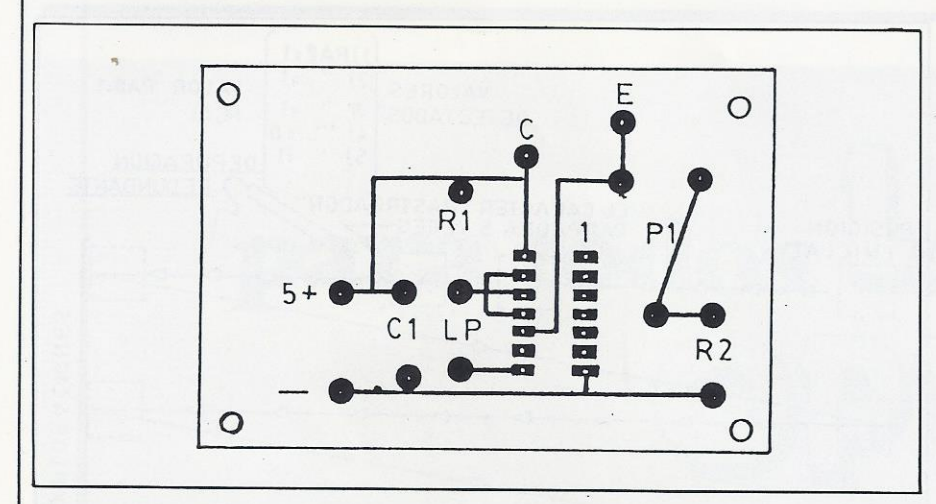


Figura 8.2. Placa de circuito impreso.

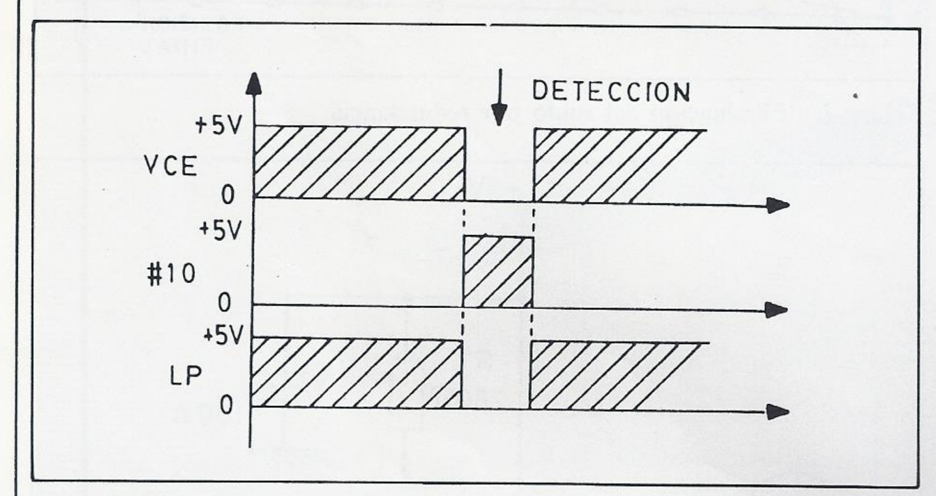


Figura 8.3. Señales.

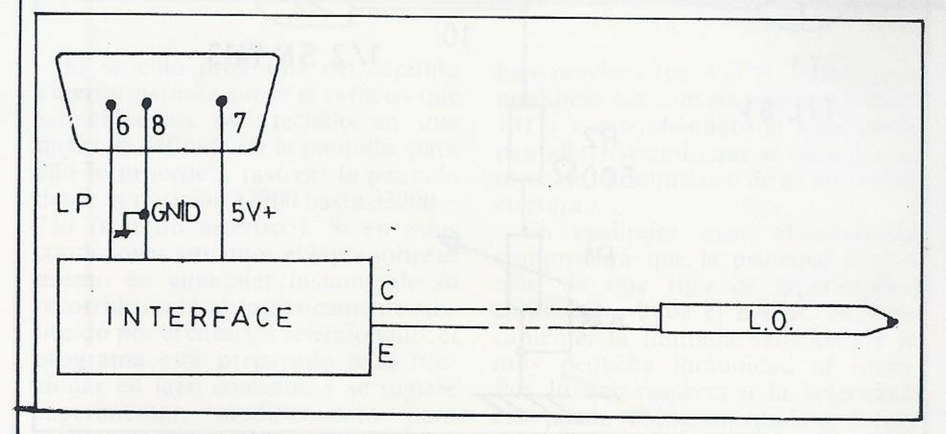


Figura 8.4. Conexión al ordenador.

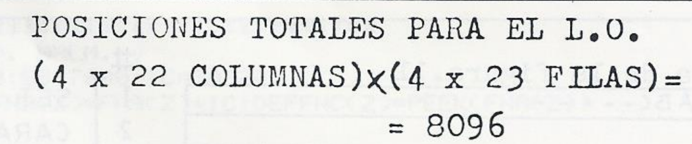
zar comprobaciones en distintas zonas de la pantalla.

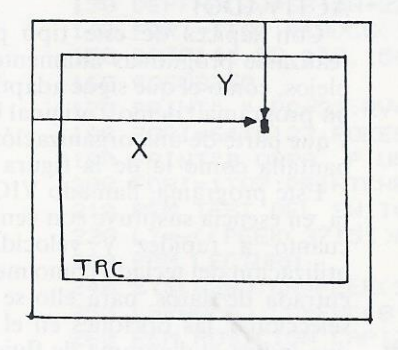
El lápiz descrito puede dar dificultades en programas en los que se juegue con caracteres de distintos colores, debido a la no igualdad de la respuesta espectral del transistor y al comprometido ajuste del basculamiento del **Trigger.** Aunque, desde luego, éste puede ser retocado variando el valor de R1; sería preferible utilizar un foto-Darlington o mejor aún, un **Trigger** de umbral ajustable, realizado con un amplificador operacional, como el que se indica en la figura 10.

Su funcionamiento es como sigue: el T1 montado para salida por colector ataca a un comparador realizado con el 3140 que toma su referencia de R3, R4 (entrada inversora) a través de su entrada no inversora (R2). La realimentación proporcionada por R5 provoca el basculamiento enérgico de la salida del operacional con pequeñas variaciones de VCE. Los diodos en antiparalelo a la entrada del 3140 actúan como protección contra sobretensiones en ésta, provocadas por iluminaciones excesivas sin disminuir la sensibilidad del dispositivo para niveles luminosos más modestos (que provocan entradas diferenciales menores que la tensión umbral de dichos diodos).

Se incluye un interruptor analógico de estado sólido en tecnología MOS (CD 4066) controlado por un pulsador que permite disponer de una salida de SENSE exenta de rebotes que indica si se desea activar el lápiz (= "1") o no (= "0"). En una realización experimental puede ser difícil el situar este pulsador sobre el lápiz óptico, en forma de que su manejo no resulte demasiado engorroso durante la exploración de la pantalla (los modelos comerciales se benefician en este aspecto de más sofisticados diseños, así los hay que se activan por contacto de la punta contra la pantalla y también por presión lateral entre el pulgar y el índice, etc.).

La conexión de este lápiz óptico por lo demás es idéntica a los anteriores, excepto por el sense que debe conectarse a la patilla # 1 de *joystick* y





REGISTROS ASOCIADOS AL V.I.C./ LP			
		(*)VARIACION	
DIRECCION	CONTENIDO	SOLO PANTALLA	ID+BORDE
36870	X	32-119	30-122
36871	Y	24-115	17-121

```
100 PX=36870:PY=PX+1
110 X=PEEK(X):Y=PEEK(Y)
120 PRINT "LAPIZ EN X-Y";X,Y
130 GOTO110
```

(*) PUEDE VARIAR SEGUN UNIDAD.

Figura 9. Organización de la pantalla para el lápiz óptico.

Figura 10. Otro posible circuito de lápiz óptico.

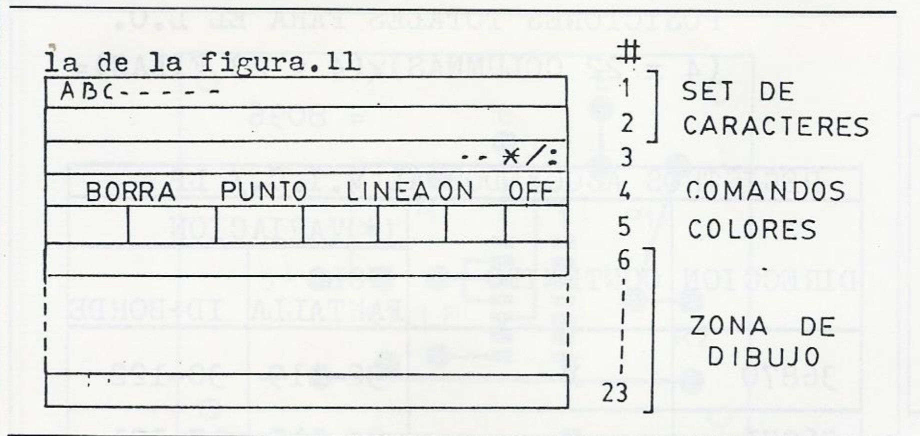


Figura 11. Formato de la pantalla para el "Vic-Paleta".

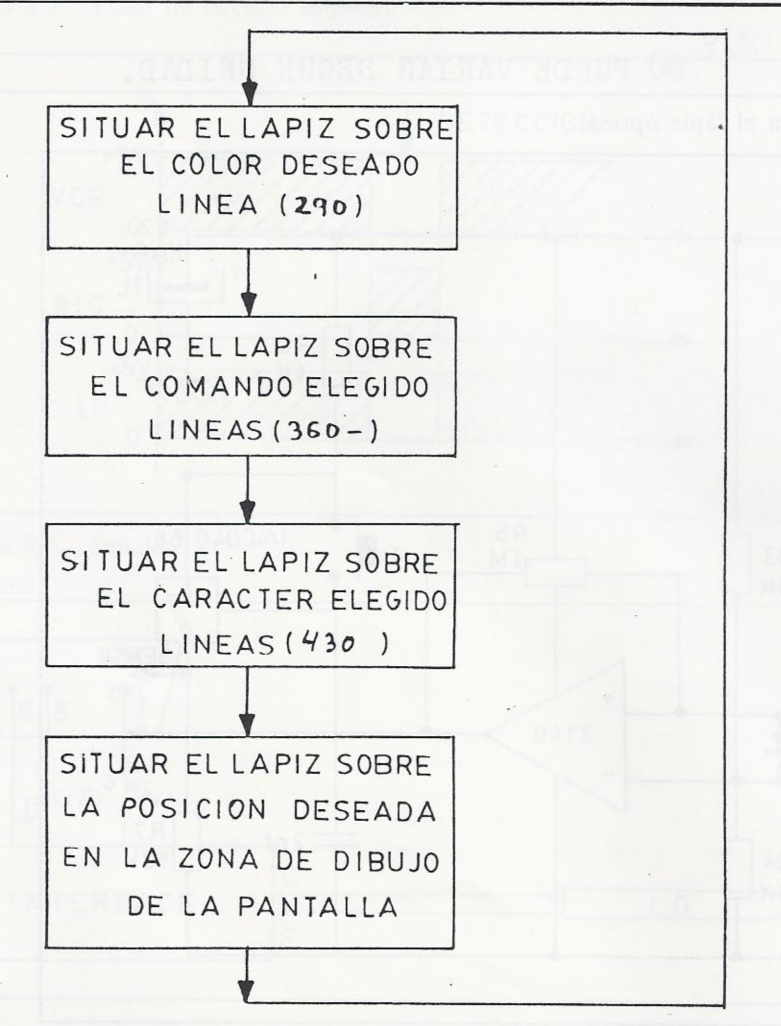


Figura 12. Diagrama de flujo de la operación con el lápiz óptico.

cuya activación puede controlarse por:

(PEEK (37151) AND 4) (= O NO ACTIVADO)

Con lápices de este tipo pueden realizarse programas altamente complejos, como el que sigue adaptado de un programa "demo" original U.S.A. y que parte de una organización de la pantalla como la de la figura 11.

Este programa, llamado VIC-Paleta, en esencia sustituye con ventaja en cuanto a rapidez y velocidad, la utilización del teclado como medio de entrada de datos, para ello se deben seleccionar las opciones en el orden que indica el diagrama de flujo de la fig. 12.

Los comandos PUNTO y LINEA permiten trazar una imagen y una línea con el punto anterior, respectivamente. Los OFF y ON permiten activar y eliminar la función "REVERSE".

El listado del programa puede verse en la fig. 13; se encuentra suficientemente aclarado con sentencias REM, por lo que sólo interesa destacar el sistema utilizado para eliminar el "ruido" por redundancia aritmética. Este sistema consiste, en esencia, en la toma de siete medidas de las lecturas del lápiz óptico al *jiffy* de intervalo, cada una; calculando posteriormente la media de las mismas.

Como quiera que la realización de este proceso sería imposible desde el BASIC en condiciones de velocidad aceptable; se incluye un programa en código máquina, que es cargado por las sentencias DATA desde 760 a 820 sobre direcciones correspondientes al *buffer* del *cassette*, que por sus características de dirección protegida resulta idónea para estos fines (¡¡importante, el uso del *cassette* puede destruir esta subrutina!!) y que se activa por SYS (828) en 240.

En la fig. 14 se incluye el listado desensamblado de la rutina en máquina por el innegable interés de su aplicación en otros programas.

```
100 REM VIC PALETA ***********************************
105 REM (ADAPTACION DE  RUT. D. MALM. )
120 DIMCC(7):POKE36879,27:CL=3:SS=7680:IC=30720
130 DEFFNA(Z)=SS+CN+22*RW:DEFFNB(Z)=FNA(Z)+IC:DEFFNC(Z)=PEEK(FNA(Z))
140 FORI=1TO7:READCC(I):NEXT
150 DATA144,28,159,156,30,31,158
160 GOSUB760
170 PRINT" ":PC=32:RV=0*LN=0:LX=0:LY=0
180 FORI=64TO127:POKESS+I-64,I:POKESS+I+IC-64,CL:NEXT
190 PRINT"B ORRA  P UNTO  L INEA  O N O FF"
200 FORI=1TO7:PRINTCHR$(CC(I))"    ";:NEXT
210 REM TEST DEL SWITCH ========================
220 LT=-((PEEK(37151)AND4)=0):IFLT=0THEN220
230 REM LECTURA          ========================
240 SYS(828):CN=PEEK(934):RW=PEEK(985)
250 L=FNA(0):LC=FNC(0)
260 REM POSICION LAPIZ  ========================
270 IFRW>4THEN520
280 IFRW<4THEN340
290 REM COLOR           ========================
300 X=INT((CN+1.1)/3):Y=(CN+1)/3
310 IFX<>YTHEN220:REM NO VALIDO ================
320 IFX=1THENX=0
330 POKE36879,24+X:CL=X:GOTO450
340 IFRW<3THEN430
350 REM COMANDOS        ========================
360 IFCN=0THEN170 :REM BORRA
370 IFCN=8THENLM=0:GOTO450 :REM PUNTO
380 IFCN=15THENLN=1:GOTO450:REM LINEA
390 IFCN=22THENRV=1:GOTO450:REM REVERSE
400 IFCN=25THENRV=0:GOTO450:REM REVERSE OFF
410 GOTO220 :REM NO VALIDO
430 PC=LC
440 REM POSICIONADO FLASH ======================
450 S=0:IFLC>127THENS=LC:LC=LC-128
460 POKE36878,15:FORI=1TO10
470 POKEL,LC+128:POKE36876,225

480 POKEL,LC:POKE36876,195
490 NEXT:IFS<>0THENPOKEL,S
500 POKE36878,0:POKE36876,0:GOTO220
510 REM RUTINAS DE DIBUJO ======================
520 V=PC:IFRV=1THENV=PC+128
530 IFLN=0THEN590
550 POKEFNA(0),V:POKEFNB(0),CL
560 LX=CN:LY=RW
570 GOTO220
580 REM LINEA DESDE EL ULTIMO PUNTO
590 BX=CN-LX:DY=RW-LY
600 Z=ABS(DX):IFABS(AY)>ZTHENZ=ABS(DY)
610 IFZ=0THENIX=0:IY=0:GOTO630
620 IX=DX/Z
630 FORI=0TOZ:RW=INT(LI+IY*I+0.5):CN=INT(LX+IX*I+0.5)
640 POKEFNA(0),V:POKEFNB(0),CL
650 NEXT:LX=CN:LI=RW:GOTO220
660 RUTINA EN C. M.      ========================
665 REM 982 = VALOR H (0-87): 983 = VALOR VERTICAL (0-91)
666 REM 984 = COLUMNA (0-21): 985 = FILA (0-22)
760 FORI=828TO986:READDC:POKEI,DC:NEXTI
```

```
770 DATA162,0,160,3,132,152,173,6,144
780 DATA201,32,176,2,169,32,201,120,144
790 DATA169,119,56,233,32,160,219,132
800 DATA151,32,179,3,165,151,24,109,218
810 DATA133,151,144,2,230,152,173,7
820 DATA144,201,24,176,2,169,24,201,116
830 DATA144,2,169,115,56,233,24,32,179
840 DATA3,232,236,218,3,240,9,165,162
850 DATA197,162,240,252,76,62,3,173,218
860 DATA3,74,168,177,151,141,215,3,169
870 DATA219,133,151,169,3,133,152,177
880 DATA151,141,214,3,173,214,3,74,74
890 DATA141,216,3,173,215,3,74,74,141
900 DATA217,3,96,142,214,3,172,214,3
910 DATA192,0,240,22,136,209,151,200
920 DATA176,16,136,141,214,3,177,151
930 DATA200,145,151,136,173,214,3,56
940 DATA176,230,145,151,96,0,0,0,0,7
```

Figura 13. Listado del programa "Vic-Paleta".

```
.,   033A FF          ???
.,   033B 11 90       ORA ($90),Y
.,   033D 1C          ???
.,   033E 9F          ???
.,   033F 9C          ???
.,   0340 1E 1F 9E    ASL $9E1F,X
.,   0343 A2 00       LDX #$00
.,   0345 A0 03       LDY #$03
.,   0347 84 98       STY $98
.,   0349 AD 06 90    LDA $9006
.,   034C C9 20       CMP #$20
.,   034E B0 02       BCS $0352
.,   0350 A9 20       LDA #$20
.,   0352 C9 78       CMP #$78
.,   0354 90 A9       BCC $02FF
.,   0356 77          ???
.,   0357 38          SEC
.,   0358 E9 20       SBC #$20
.,   035A A0 DB       LDY #$DB
.,   035C 84 97       STY $97
.,   035E 20 B3 03    JSR $03B3
.,   0361 A5 97       LDA $97
.,   0363 18          CLC
.,   0364 6D DA 85    ADC $85DA
.,   0367 97          ???
.,   0368 90 02       BCC $036C
.,   036A E6 98       INC $98
.,   036C AD 07 90    LDA $9007
.,   036F C9 18       CMP #$18
.,   0371 B0 02       BCS $0375
.,   0373 A9 18       LDA #$18
.,   0375 C9 74       CMP #$74
.,   0377 90 02       BCC $037B
.,   0379 A9 73       LDA #$73
.,   037B 38          SEC
.,   037C E9 18       SBC #$18
.,   037E 20 B3 03    JSR $03B3
.,   0381 E8          INX
.,   0382 EC DA 03    CPX $03DA
.,   0385 F0 09       BEQ $0390
..   0387 A5 A2       LDA $A2
```

```
.,   0389 C5 A2       CMP $A2
.,   038B F0 FC       BEQ $0389
.,   038D 4C 3E 03    JMP $033E
.,   0390 AD DA 03    LDA $03DA
.,   0393 4A          LSR
.,   0394 A8          TAY
.,   0395 B1 97       LDA ($97),Y
.,   0397 8D D7 03    STA $03D7
.,   039A A9 DB       LDA #$DB
.,   039C 85 97       STA $97
.,   039E A9 03       LDA #$03
.,   03A0 85 98       STA $98
.,   03A2 B1 97       LDA ($97),Y
.,   03A4 8D D6 03    STA $03D6
.,   03A7 AD D6 03    LDA $03D6
.,   03AA 4A          LSR
.,   03AB 4A          LSR
.,   03AC 8D D8 03    STA $03D8
.,   03AF AD D7 03    LDA $03D7
.,   03B2 4A          LSR
.,   03B3 4A          LSR
.,   03B4 8D D9 03    STA $03D9
.,   03B7 60          RTS
.,   03B8 8E D6 03    STX $03D6
.,   03BB AC D6 03    LDY $03D6
.,   03BE C0 00       CPY #$00
.,   03C0 F0 16       BEQ $03D8
.,   03C2 88          DEY
.,   03C3 D1 97       CMP ($97),Y
.,   03C5 C8          INY
.,   03C6 B0 10       BCS $03D8
.,   03C8 88          DEY
.,   03C9 8D D6 03    STA $03D6
.,   03CC B1 97       LDA ($97),Y
.,   03CE C8          INY
.,   03CF 91 97       STA ($97),Y
.,   03D1 88          DEY
.,   03D2 AD D6 03    LDA $03D6
.,   03D5 38          SEC
.,   03D6 B0 E6       BCS $03BE
.,   03D8 91 97       STA ($97),Y
.,   03DA 60          RTS
```

Figura 14. Listado del desensamblado de la subrutina en código máquina.

SU PROGRAMA PARA CUALQUIER SISTEMA COMMODORE PUEDE HACERLE GANAR 5.000 PTAS.

EL PRESENTE CONCURSO ESTA ABIERTO A TODOS NUESTROS LECTORES Y SU PARTICIPACION E INSCRIPCION ES GRATUITA. LEA LAS BASES DEL CONCURSO

- NO SE ESTABLECEN LIMITACIONES EN CUANTO A EXTENSION, TEMA ELEGIDO O MODELO DE ORDENADOR
- LOS CONCURSANTES DEBERAN ENVIARNOS A LA DIRECCION QUE FIGURA AL PIE, EL CASSETTE O DISKETTE CONTENIENDO EL PROGRAMA, UNA EXPLICACION DEL MISMO Y, AL SER POSIBLE, UN LISTADO EN PAPEL DE IMPRESORA. SE PODRAN ENVIAR TANTOS PROGRAMAS COMO SE DESEE
- LOS PROGRAMAS, PREVIA SELECCION, SERAN PUBLICADOS EN LA REVISTA, OBTENIENDO TODOS ELLOS 5.000 PTAS.
- LA DECISION SOBRE LA PUBLICACION O NO DE UN PROGRAMA CORRESPONDE UNICAMENTE AL JURADO NOMBRADO AL EFECTO POR "COMMODORE MAGAZINE", SIENDO SU FALLO INAPELABLE
- LOS CRITERIOS DE SELECCION SE BASARAN EN LA CREATIVIDAD DEL TEMA ELEGIDO Y LA ORIGINALIDAD Y/O SENCILLEZ EN EL METODO DE PROGRAMACION GLOBAL
- ENVIAR A: CONCURSO COMMODORE MAGAZINE

ORIGINALIDAD

Los programas han de ser inéditos. No deben haberse enviado a otras publicaciones, ni ser copiados de manuales y/o libros, ya sean españoles o extranjeros.



commodore Magazine

C/Bravo Murillo, 377. 28002-Madrid -

Ampliación de memoria

Es posible dotar al **Vic-20** de una capacidad de memoria superior a los 5 Kbytes de RAM con los que viene de fábrica. Para ello **Commodore** ha dispuesto en la parte posterior de la máquina un *slot* de expansión del sistema.

Existen una serie de cartuchos de ampliación de la memoria central, enchufables en el *slot* antes aludido. Sus principales fuentes de procedencia son el propio fabricante de la marca o terceros.

En el presente artículo presentamos un circuito de ampliación de RAM, que proporciona al **Vic-20** otros 3 Kbytes adicionales. Son de agradecer, puesto que la máquina viene con sólo 5 Kbytes de RAM, de los cuales sólo 3,5 K son utilizables por el usuario. Su distribución es la siguiente: 1 Kbyte situado entre las direcciones $0000 y $03FF (hexadecimales) y los siguientes 4 Kbytes se distribuyen entre las $1000 y $1FFF, pero hay una zona de 3 K, situada entre las $0400 y $0FFF, que no tiene nada asignado.

Es posible aprovechar una ventaja de diseño que tiene el **Vic-20** para poder direccionar bloques de memoria de 1 Kbyte cada uno. Existe una señal llamada CS en los *chips* de memoria, que responde al nombre de *Chip Select* (selección de un determinado *chip*). Cuando esta señal es un cero lógico, el microprocesador se está dirigiendo a ese *chip* en particular y no a otro. En el *slot* de expansión del ordenador existen 3 señales, etiquetadas cada una como RAM1, RAM2 y RAM3, correspondientes a las patillas 14, 15 y 16 respectivamente. En realidad se trata de señales aplicables directamente al *Chip Select* del módulo de memoria. Simplemente con estas 3 señales es posible ampliar la memoria del **Vic-20** hasta 8 Kbytes más, sin necesidad de realizar ninguna decodificación.

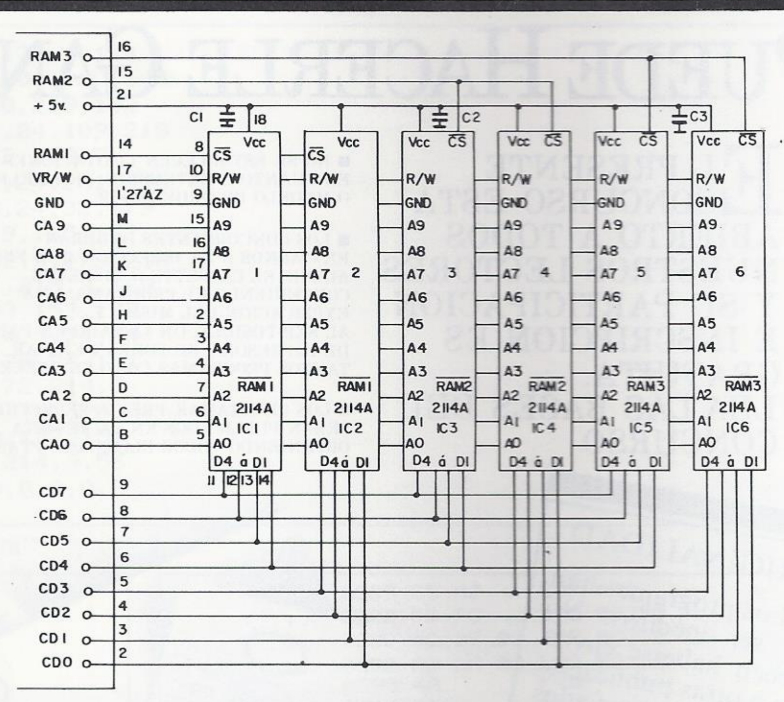


Figura 1. Esquema teórico del circuito.

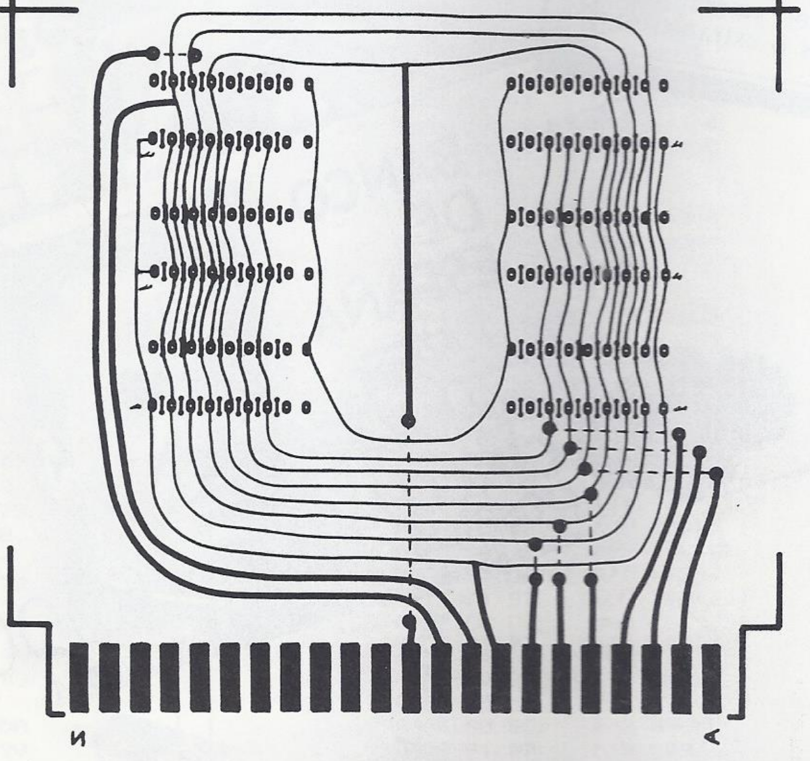


Figura 2. Cara inferior del circuito impreso. Los trazos discontinuos representan los puentes de hilo metálico.

El circuito

Los *chips* adoptados para este montaje son de corriente utilización en diversos modelos y marcas de ordenador. Estamos refiriéndonos al 2114, que además es el que utiliza el

para el VIC-20

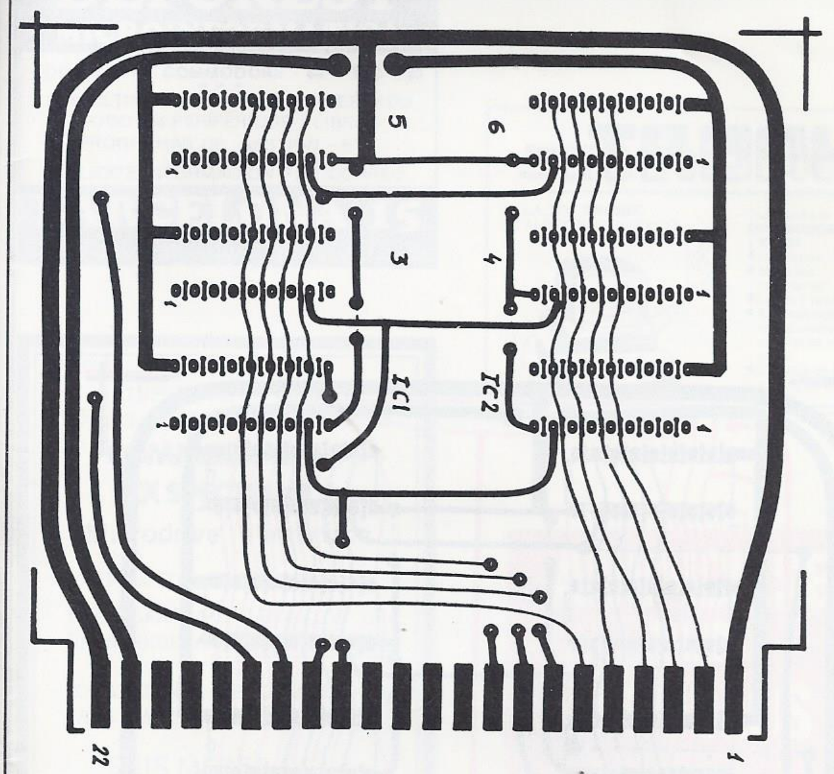


Figura 3. Cara superior del circuito impreso (vista desde el lado de los componentes). Los trazos discontinuos son los puentes que se realizan después con hilo metálico.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

A B C D E F H J K L M N P R S T U V W X Y Z

Contacto	TIPO
1	TIERRA
2	CDØ
3	CD1
4	CD2
5	CD3
6	CD4
7	CD5
8	CD6
9	CD7
1Ø	BLK1
11	BLK2

Contacto	TIPO
12	BLK3
13	BLK5
14	RAM1
15	RAM2
16	RAM3
17	VR/W
18	CR/W
19	IRQ
2Ø	NC
21	+5V
22	TIERRA

Contacto	TIPO
A	TIERRA
B	CAØ
C	CA1
D	CA2
E	CA3
F	CA4
H	CA5
J	CA6
K	CA7
L	CA8
M	CA9

Contacto	TIPO
N	CA1Ø
P	CA11
R	CA12
S	CA13
T	I/Ø2
U	I/Ø3
V	SØ2
W	NMI
X	RESET
Y	NC
Z	TIERRA

Figura 4. *Slot* de expansión del **Vic-20** con su tabla de correspondencias.

Vic-20, en su memoria interna. Con la ayuda de unos cuantos de estos *chips* y las señales RAM1, RAM2 y RAM3 obtenemos la expansión.

Por su diseño interno, el 2114 puede almacenar hasta 1024 *nibbles,* que no son ni más ni menos que medio byte cada uno o, si lo preferimos, 4 bits. Son memorias estáticas (por contra de dinámicas) de bajo consumo. El *chip* va encapsulado en una pastilla tipo DIL (doble hilera de patillas) con 18 terminales, que se distribuyen como sigue: A0 a A9 (diez patillas), son líneas de direccionamiento. Los datos leídos o escritos de la memoria lo son a partir de las patillas D0 a D3, llamadas de datos, siendo 4 en este caso. La patilla R/W hace saber al *chip* si el microprocesador quiere leer datos a partir de la memoria o depositarlos en ella (cuando está a cero lógico).

Un byte está compuesto por dos *nibbles,* por tanto necesitamos dos *chips* capaces de almacenas 1024 *nibbles* para obtener 1 Kbyte de almacenamiento. Los *chips* van emparejados dos a dos, por tanto seis proporcionan los 3 Kbytes de memoria requeridos.

Si observamos el esquema teórico, las patillas CS de cada dos *chips* van unidas entre sí y a las salidas RAM1, RAM2 y RAM3 respectivamente. Las líneas de datos de cada uno de los grupos de dos van conectadas alternativamente al *nibble* más significativo y al menos significativo del bus de datos del ordenador.

Las patillas R/W (lectura/escritura) de todos los *chips* están unidas entre sí, pues sólo estarán activos en un momento dado los que tengan activado su CS, ignorando el resto cualquier otro tipo de señal mientras tanto.

Los *chips* de memoria reciben su alimentación desde el **Vic-20,** a través de los terminales +5V y *Ground* (masa).

Montaje

De acuerdo con el esquema de la figura 1, en las figuras 2 y 3 aparece el diseño del circuito impreso correspondiente. Se trata por tanto de un circuito impreso de doble cara.

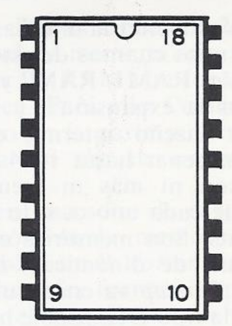


Figura 5. Diagrama de patillas de una RAM estática 2114.

Cuando se ha terminado de preparar la tarjeta de circuito impreso, es importante efectuar los puentes de hilo necesarios para asegurar algunas conexiones eléctricas. En las figuras van indicados por trazos discontinuos.

El siguiente paso consiste en insertar y conectar las patillas de los zócalos que soportarán a los *chips*. En este tipo de montaje es altamente recomendable utilizarlos, pues de otro modo es bastante probable que los resulten dañados durante el proceso de la soldadura, a no ser que uno conozca la técnica adecuada. Habrá que tomar la precaución de que las patillas sean soldadas por ambas pistas de la tarjeta.

Por último se soldarán los condensadores de desacoplo, que si bien no son de utilización obligatoria si son muy recomendables. Igualmente habrá que tener cuidado de no insertar los *chips* en posición errónea.

Si los puentes han sido bien puestos y las soldaduras efectuadas correctamente, el montaje trabajará perfectamente desde el principio sin necesidad de ajustes.

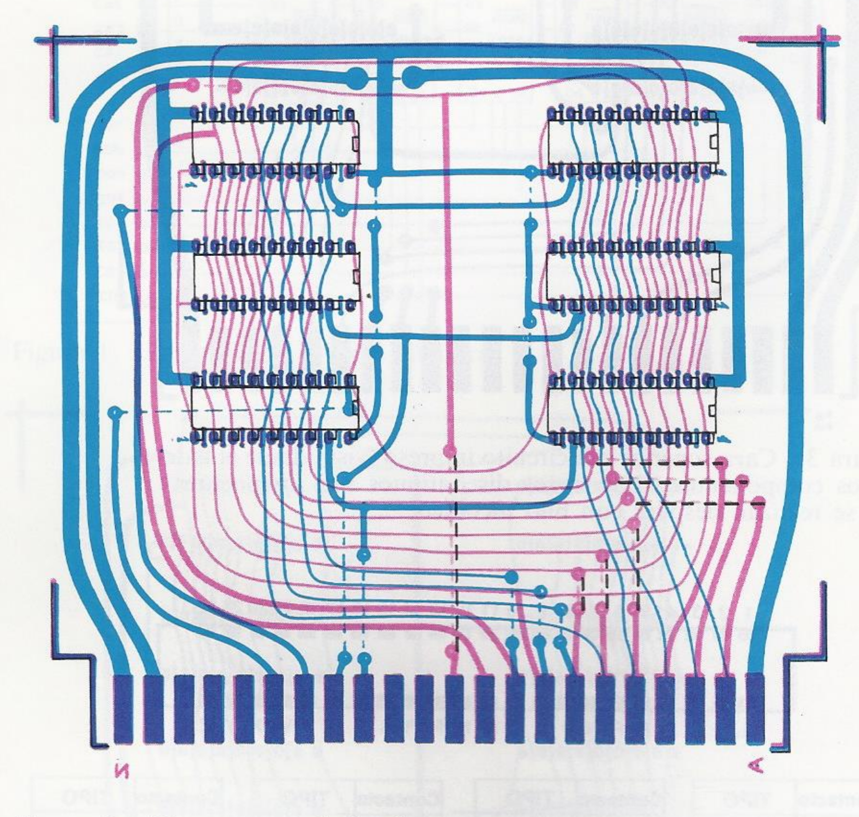


Figura 6. Vista general de la placa terminada.

GUIA PRACTICA



Programas

Misión suicida

Por una serie de desafortunadas circustancuas, tu nave, averiada y sin apenas combustible, ya no puede regresar a la tierra. El fin es seguro, pero no importa, lo importante es, antes de que llegue el momento, acabar con el mayor número de marcianitos posible. Por todo ello has decidido embarcarte en esta "misión suicida" mientras funcionen tus misiles y hasta que no quede ni una gota de combustible. Los marcianitos no van a faltar. Intentarán destruirte con sus disparos que debes evitar. Sólo en el caso de que te vayan a acertar sin remedio debes utilizar la coraza de protección de que dispones, ya que consume mucho combustible. La cantidad que te queda de éste, podrás verla en un indicador en la parte derecha de la pantalla. Cuando termines todo habrá acabado.

Los controles de tu nave, todavía en buen estado, son:

W = Arriba
X = Abajo
A = Izquierda
D = Derecha
S =Misiles
Espacio = Coraza de protección

La suerte está echada, sólo te queda el consuelo de acabar habiéndote llevado por delante el mayor número de marcianitos posible, y ten por seguro que tus hazañas serán recordadas por mucho tiempo allá abajo, en tu planeta natal, la tierra.

Lo que sigue es una lista de las variables utilizadas por el programa que pueden servir para posibles modificaciones que se le ocurran al lector:

PE = última tecla pulsada
F = Fuel
SX, SY = Posiciones horizontal y vertical del misil del jugador
P = Posición horizontal del jugador
YP = Posición vertical del jugador
X, Y = Posiciones horizontal y vertical del enemigo
EX, EY = Posiciones horizontal y vertical del misil enemigo

```
10 REM ************************
15 REM *  MISION  SUICIDA     *
20 REM *                      *
25 REM * COMMODORE MAGAZINE *
30 REM ************************
35 POKE55,255:POKE56,47:PRINT"♥":POKE53281,0:POKE53280,0:V=53248:CO=54272
40 SM=0:FORJ=1TO30:I=RND(1)*1000+55296:POKEI,1:POKEI-CO,46:NEXT
45 POKEV+21,31:POKE2040,192:POKE2041,193:POKE2042,194:POKE2043,195:POKE2044,196
50 POKEV+39,8:POKEV+40,3:POKEV+41,7:POKEV+42,10:POKEV+43,5
55 FORI=0TO318:READQ:POKE12288+I,Q:NEXT:POKE54296,1:POKE54278,240:POKE54276,129
60 P=120:YP=200:POKEV,P:POKEV+1,YP:X=50+INT(RND(1)*180):Y=0:POKEV+2,X:POKEV+3,Y
65 PE=PEEK(V+30):F=0:POKE54272,108:POKE54273,223
70 PRINT"[control chars]PUNTOS:":POKEV+4,200
75 PRINT"[control chars]"
80 PE=PEEK(197):F=F+.01:POKEV+39,8:IFHH<>0THENPOKEV+39,12
85 PRINT"[control chars]";S
90 POKE55613+INT(F)*40,4
95 IFPE=10ANDP>50THENP=P-5:F=F+.01
100 IFPE=18ANDP<239THENP=P+5:F=F+.01
105 IFPE=9ANDYP>175THENYP=YP-5:F=F+.01
110 IFPE=23ANDYP<230THENYP=YP+5:F=F+.01
115 IFPE=13ANDSH=0THENGOSUB410
120 IFPE=60ANDHH=0THENHH=1:F=F+1
125 IFSH=0THEN150
130 FORI=1TO2:SY=SY-SH:POKEV+5,SY
135 IFPEEK(V+30)=6THENGOSUB305
140 IFSY<60THENSY=0:SH=0:SX=150:POKEV+5,0
145 NEXT
150 IFHH<>0THENHH=HH+1:IFHH=10THENHH=0
155 POKEV,P:POKEV+1,YP:IFF>15THEN260
160 IFRND(1)<.1THENRN=INT(RND(1)*3-1):IFRN=0ANDRND(1)<.1THEN160
165 IFX>239THENRN=-1
```

```
170 X=X+RN*4:IFX<50THENRN=1
175 Y=Y+4:IFY>235THENY=0:X=50+INT(RND(1)*180):SM=SM+1
180 POKEV+2,X:POKEV+3,Y
185 IFEY=0ANDRND(1)<.1 ANDY<220THENEY=Y+20:EX=X:POKEV+6,X:GOSUB430
190 IFEY=0THENGOTO80
195 FORI=1TO2:EY=EY+6:POKEV+7,EY:POKEV+6,EX
200 IFRND(1)<.2THENEX=EX+((EX>P)-(EX<P))*5
205 IFEX<35THENEX=35
210 IFEX>238THENEX=238
215 IFEY>235THENEY=0:EX=0:POKEV+7,EY:POKEV+6,EX:GOTO80
220 IFPEEK(V+30)=9ANDHH=0THEN335
225 NEXT:IFPEEK(V+30)<>12THEN80
230 POKE54273,0:POKE54272,0:POKE54296,5:FORI=30TO1STEP-1:POKE54273,I:NEXT
235 POKE54296,1:  :POKE54273,223:POKE54272,108
240 POKEV+5,0:POKEV+7,0:POKEV+8,EX:POKEV+9,EY
245 FORI=1TO100:NEXT
250 EX=50:EY=0:SY=0:SY=O:SX=0:SH=0:POKEV+8,0:POKEV+9,0
255 GOTO80
260 GOSUB375
265 PRINT"■■TE HAS QUEDADO SIN ENERGIA!"
270 PRINT"■■■MALA SUERTE, AUNQUE HAS AGUANTADO MUCHO!"
275 PRINT"■■■■■■■■■■■■■■■■■■---------------------------------------------"
```

Viene de la página anterior

```
280 PRINT"▒▒PARA JUGAR OTRA VEZ PULSA 1"
285 GETA$:IFA$="1" THEN RUN
290 IFA$<>"1"THEN285
305 POKE54272,0:POKE54296,10:FORI=50TO1STEP-1.5:POKE54273,I:NEXT
310 POKE54296,1:POKE54273,223:POKE54272,108
315 POKEV+2,0:POKEV+3,0:POKEV+8,X:POKEV+9,Y
320 S=S+10:IFY=SYTHENS=S+40
325 FORZ=1TO100:NEXT:POKEV+8,0:POKEV+9,0
330 SH=0:SY=0:SX=0:Y=0:RETURN
335 GOSUB375
340 PRINT"▒HAS SIDO ALCANZADO POR UN MISIL ENEMIGO!"
345 PRINT"▒▒▒ MALA SUERTE"
350 PRINT"▒▒▒PUNTOS:";S
355 PRINT"UNIDADES DE ENERGIA GASTADA";INT(F)
360 PRINT"▒NAVES ENEMIGAS NO ALCANZADAS";SM
365 PRINT"▒▒▒NO ESTA MAL!"
370 GOTO275
375 FORI=1TO7:POKEV+I,0:NEXT
380 POKE54273,0:POKE54272,0:POKE54276,0
385 POKE54296,15:POKE54276,129
390 POKEV+8,P:POKEV+9,YP
395 FORI=50TO1STEP-1:POKE54273,I:NEXT:POKE54273,0:POKE54272,0:POKE54296,1
400 POKEV+8,0:POKEV+9,0
405 RETURN
410 POKE53280,2:SH=6:SX=P:SY=YP:POKEV+4,SX:F=F+.05:POKE53280,0
415 POKE54273,0:POKE54272,0:POKE54296,10:FORI=0TO150STEP3:POKE54273,I:NEXT
420 POKE54273,0:POKE54296,1
425 RETURN
430 POKE54273,0:POKE54272,0:POKE54276,0:POKE54276,33
435 POKE54273,50:FORZX=1TO50:NEXT:POKE54276,129:POKE54273,233:POKE54272,108
440 RETURN
445 DATA0,0,0,8,0,0,28,0,0,28,0,0,62,0,0,127,0,0,127,0,0,127,0,0,127,0,2
450 DATA127,32,2,28,32,7,28,112,2,62,32,6,255,176,7,235,240,7,235,240,15,255
455 DATA248,15,62,120,28,28,60,24,0,12,0,0,0,99
460 DATA0,130,0,64,130,4,112,214,28,120,154,60,124,124,124,62,56,248,31,255
465 DATA240,15,255,224,7,255,192,1,255,0,0,254,0,12,108,96,6,108,192,7,255,192
470 DATA3,255,128,1,255,0,0,254,0,0,56,0,0,16,0,0,16,0,0,16,0,99
475 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0
480 DATA32,7,0,112,15,128,248,31,193,252,26,192,172,18,65,36,2,0,32,0,0,0
485 DATA0,0,0,2,0,32,99
490 DATA0,0,0,0,0,0,0,0,0,0,1,215,0,0,254,0,0,254,0,0,124,0,0,16,0,0
495 DATA16,0,0,16,0,0,124,0,0,56,0,0,16,0,0,16,0,0,0,0,0,0,0,0,0,0,0,0,0,0
500 DATA0,0,0,0,0,0,99
505 DATA3,12,0,0,3,0,192,192,195,3,204,192,12,3,12,0,12,48,0,51,3,48,51,0
510 DATA3,12,0,0,3,0,192,192,195,3,204,192,12,3,12,0,12,48,0,51,3,48,51,0
515 DATA3,12,0,0,3,0,192,192,195,3,204,195,3,204,195,3,204
```

Tanque destructor

CBM - 64

Una serie de naves de no se sabe qué rincón de la galaxia están intentando aterrizar, una por una, precisamente en la zona en la que tú estás de guardia a bordo de tu tanque destructor. Tu misión es, obviamente, destruir las naves antes de que puedan aterrizar, ya que si lo hacen soltarán su carga de seres de otros mundos antes de desaparecer de nuevo en la negrura del espacio. Para destruirlas, tu tanque dispone de un

láser que podrás disparar cuantas veces quieras, con sólo pulsar la tecla *"M"*. Para desplazar tu tanque por la superficie del terreno puedes emplear las teclas Z (para ir hacia la izquierda) y C (para ir hacia la derecha). Debes tener en cuenta, además, que las naves invasoras disponen de campos energéticos de protección que utilizarán en determinados momentos. Si la nave aparece de color azul, significa que no tiene conectado su campo, por lo que puede ser derribada. Si por el contrario aparece de color rojo, ni el más certero disparo podrá ni siquiera rayar la superficie metálica de l nave. Aparte de este campo de protección, las naves invasoras también disponen de un láser con el que intentarán destruirte; sus disparos son bastante certeros, y las únicas zonas un poco seguras son las esquinas de la pantalla.

Las variables que utiliza el programa para aquellos interesados en destriparlo o modificarlo son las siguientes:
P=Posición del tanque.
A=Posición de los alienígenas (marcianitos).
PE=Tecla pulsada.
B=Posición de la bomba láser.

```
10 REM ************************
15 REM * TANQUE  DESTRUCTOR *
20 REM *                    *
25 REM * COMMODORE MAGAZINE *
30 REM ************************
35 POKEV+55,255:POKEV+56,47:PRINT"J":POKE53281,0:POKE53280,0
40 V=53248:POKE2040,192:POKE2041,194:POKE2042,196:POKE2043,197:POKE2044,198
45 POKEV+39,3:POKEV+40,2:POKEV+41,2:POKEV+42,3:POKEV+43,12:POKEV+28,15
50 POKEV+37,8:POKEV+38,5
55 FORI=0TO446:READQ:POKE12288+I,Q:NEXT
60 FORI=56136TO56295:POKEI,9:POKEI-54272,160:NEXT:POKEV+21,31:POKE54273,0
```

Programas

Viene de la página anterior

```
65 POKE54272,0:POKE54276,0:POKE54277,0:POKE54278,0:POKE54296,15:POKE54278,240
70 POKE54276,129:POKEV+23,0:POKEV+29,4:P=110:POKEV+1,200:POKEV+2,11
75 POKEV+5,50:POKEV+4,230:A=150+INT(RND(1)*70):B=50:PE=PEEK(V+30):G=.4
80 POKEV+8,0:POKEV+9,220
85 POKE54273,0:PE=PEEK(V+30)
90 PE=PEEK(197)
95 IFPE=12ANDP>30THENP=P-3:POKE2040,192:POKE54273,2
100 IFPE=20ANDPE<320THENP=P+3:POKE2040,193:POKE54273,2
105 IFP>255THENPOKEV,P-256:POKEV+16,PEEK(V+16)OR1:GOTO115
110 POKEV,P:POKEV+16,PEEK(V+16)AND254
115 IFPE<>36THEN175
120 FORI=50TO0STEP-2:POKE54273,I:NEXT
125 X=P-5:Y=192:D=-1:POKE2041,195:IFPEEK(2040)=193THEND=1:X=X+20:POKE2041,194
130 IFPEEK(V+30)<>6THEN150
135 POKEV+2,0:POKEV+3,190:X=30:FORI=0TO10:POKE54273,I:NEXT
140 IFPEEK(V+41)=246THENGOSUB295
145 POKE54273,0
150 IFX<40ORX>300ORY<50THENX=0:Y=0:POKEV+2,0:POKEV+3,192:POKE54273,0:GOTO175
155 X=X+D*5:Y=Y-6
160 POKEV+3,Y:IFX>255THENPOKEV+2,X-256:POKEV+16,PEEK(V+16)OR2:GOTO170
165 POKEV+2,X:POKEV+16,PEEK(V+16)AND253
170 GOTO130
175 G=G+.02:IFRND(1)<.6THENA=A-3
180 B=B+G:IFRND(1)<.1THENA=A+4
185 IFA>320 THENA=240
190 IFA<40THENA=40
195 IFA>255THENPOKEV+4,A-256:POKEV+16,PEEK(V+16)OR4:GOTO205
200 POKEV+4,A:POKEV+16,PEEK(V+16)AND251
205 POKEV+5,B:IFRND(1)<.3THENPOKEV+41,2
210 IFRND(1)<.1THEN POKEV+41,6
215 IFB>185THEN350
220 IFRND(1)<.96THENGOTO85
225 IFA<PTHENPOKE2041,195
230 IFA>PTHENPOKE2041,194
235 FORI=0TO40STEP.4:POKE54273,I:NEXT:POKE54273,0
240 SH=A:FORI=BTO215STEP6
245 IFRND(1)<.4THENSH=SH+((SH>P)-(SH<P))*5
250 IFPEEK(V+30)=3THEN430
255 IFSH>255THENPOKEV+2,SH-256:POKEV+16,PEEK(V+16)OR2:GOTO265
260 POKEV+2,SH:POKEV+16,PEEK(V+16)AND253
265 POKEV+3,I
270 NEXT
275 POKEV+2,0:POKEV+3,198
280 GOTO85
285 A=INT(RND(1)*50)+170:B=50:X=30:POKEV+29,4:POKEV+23,4
290 POKEV+16,PEEK(V+16)AND251
295 POKEV+2,0:POKEV+3,192
300 POKE54273,(B-40)
305 G=G+.02:B=B+G:POKEV+5,B:IFB<205THEN300
310 POKE54273,2:POKEV+29,20:POKEV+9,B-5:POKEV+4,0:POKEV+5,0
315 IFA>255THENPOKEV+8,A-256:POKEV+16,PEEK(V+16)OR16:GOTO325
320 POKEV+8,A:POKEV+16,PEEK(V+16)AND239
325 FORZ=10TO0STEP-.2:POKE54273,I:NEXT:A=150+INT(RND(1)*70):B=50:POKE54273,0
330 FORZ=1TO500:NEXT:POKEV+29,4:POKEV+8,0:POKEV+9,220
335 IFA>255THENPOKEV+4,A-256:POKEV+16,PEEK(V+16)OR4:GOTO345
340 POKEV+4,A:POKEV+16,PEEK(V+16)AND 251
345 POKEV+5,B:G=0:RETURN
```

```
350 POKEV,0:POKEV+1,0:POKEV+2,0:POKEV+3,0:POKE54276,0:POKE54276,17
355 POKEV+8,0:POKEV+9,0
360 PRINT"[illegible]LOS MARCIANOS TE ESTAN INVADIENDO"
365 A=A+10:FORT=1TO3:IFA>255THENPOKEV+6,A-256:POKEV+16,PEEK(V+16)OR8:GOTO375
370 POKEV+6,A:  :POKEV+16,PEEK(V+16)AND247
375 FORI=BTO205:POKEV+7,I:POKE54273,I-185:NEXTI
380 FORI=ATO0STEP-1
385 POKEV+6,I:POKEV+16,PEEK(V+16)AND 247
390 POKE54273,I:NEXTI,T
395 FORI=BTO20STEP-1:POKEV+5,I:POKE54273,INT(RND(1)*200):NEXT
400 POKE 54273,0:POKE54276,0:POKE54276,33
405 FORI=50TO1STEP-.4:POKE54273,I:NEXT:POKE54273,0
410 PRINT"[illegible]JUEGAS OTRA VEZ? S/N"
415 GETA$:IFA$="S"THENRUN
420 IFA$<>"N"THEN415
425 END
430 POKE54276,0:POKE54276,129:POKEV+2,0:POKEV+3,190
435 POKEV,0:POKEV+1,0:POKEV+9,200
440 IFP>255THENPOKEV+8,P-256:POKEV+16,PEEK(V+16)OR16:GOTO450
445 POKEV+8,P:POKEV+16,PEEK(V+16)AND233
450 FORI=20TO0STEP-1:POKE54273,I:NEXT:POKE54273,0
455 POKE54276,0:POKE54276,33
460 FORI=200TO0STEP-2:POKE54273,I:NEXT
465 FORI=0TO10STEP.2:POKE54273,I:NEXT
470 POKE54273,0:POKE54276,0:POKE54276,17
475 FORI=10TO30STEP.1:POKE54273,I:NEXT
```

PROGRAMA ★ commodore Magazine ★ COMPROBADO

```
480 FORI=50TO0STEP-.4:POKE54273,I:NEXT
485 POKE54273,0
490 B=B+1:POKEV+5,B:POKE54273,B:IFB<185THEN490
495 POKEV,0:POKEV+1,0:POKE54273,0
500 GOTO 350
505 DATA0,0,0,0,0,0,0,0,0,2,0,0,0,128,0,0,128,0,0,32,0,0,8,8,0,9,8,0,21,8,0
510 DATA21,8,0,85,72,0,85,72,3,255,248,15,255,252,51,51,51,51,51,51,15,255,252
515 DATA3,255,240,0,0,0,0,0,0,99
520 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,32,0,0,128,0,0,128,0,2,0,8,9,0,8,21,0,8
525 DATA21,0,8,85,64,8,85,64,11,255,240,15,255,252,51,51,51,51,51,51,15,255
530 DATA252,3,255,240,0,0,0,0,0,0,99
535 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
540 DATA0,10,0,0,38,0,0,38,0,0,38,0,0,40,0,0,32,0,0,128,0
545 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,99
550 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
555 DATA0,160,0,0,152,0,0,152,0,0,152,0,0,40,0,0,8,0,0,2,0
560 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,99
565 DATA0,48,0,0,48,0,0,252,0,0,48,0,3,87,0,9,101,128,252,100,124,61,101,240
570 DATA15,155,192,3,255,0,1,253,0,1,49,0,4,0,64,16,0,16,136,0,136,0,0,0,0,0
575 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,99
580 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
585 DATA0,4,0,0,29,0,0,29,0,0,4,0,0,4,0,0,34,0,0,34,0
590 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,99
595 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
600 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
605 DATA0,0,0,0
610 DATA3,7,0,7,223,184,15,255,248,15,255,252,63,255,255,255,255,255
615 DATA255,255,255,255,255,255,255,255,255
```

Programas

Duelo en las estrellas

CBM 64

El malvado *RAM* te reta a un duelo a muerte sobre la superficie de su inhóspito planeta. Todo depende de ti ya que si aceptas su desafío tendrás que luchar con tu nave contra la suya, disparando tus misiles hasta que consigas alcanzarle y evitando al mismo tiempo sus disparos y las agudas crestas de los montes de la superficie del planeta, que destrozarán el casco de tu nave si no los evitas. Tanto tu nave como la del malvado *RAM* pueden moverse libremente por la pantalla. Tu controlarás el movimiento de tu nave mediante las siguientes teclas:

W = Arriba
X = Abajo
A =Izquierda
D = Derecha

La nave de tu oponente viene controlada por el programa.

Si cualquiera de las dos naves es alcanzada por un misil, quedará fuera de combate y se estrellará sin remedio contra la superficie del planeta.

Algunas de las variables utilizadas por el programa son:

X, Y = Tu posición
X1, Y1 =Posición de tu adversario
D = Tu dirección
D1 = Dirección de tu adversario
SX = Posición de tu misil
SD = Dirección de tu misil
SZ = Posición del misil enemigo
DZ = Dirección del misil enemigo

Ahora, en tus manos está la elección sobre si te enfrentas o no a este insolente adversario, en cualquier caso te deseamos buena suerte.

```
10 REM *****************************
15 REM * DUELO EN LAS ESTRELLAS *
20 REM *                           *
25 REM *   COMMODORE MAGAZINE    *
30 REM *****************************
35 :
40 POKE 54296,10:POKE54277,0:POKE54278,240:POKE54276,0:POKE54276,129:SX=0
45 POKE54272,0:POKE54273,0:PRINT"♥":POKE53280,4:POKE53281,4:CO=54272
50 V=53248:POKE2040,192:POKEV+39,14:POKE2041,193:POKE2042,194:POKE2043,194
55 POKE2044,195:POKEV+40,5:POKEV+43,0:POKEV+28,31:POKEV+37,2:POKEV+38,7
60 FORI=0TO254:READQ:POKE12288+I,Q:NEXT:POKEV+21,31:POKEV+29,19:POKEV,100
65 POKEV+2,200:POKEV+4,50:POKEV+6,230:POKEV+8,165
70 GOSUB380
75 PE=PEEK(V+30):PE=PEEK(V+31):X=100:Y=60:D=4:X1=200:Y1=120:D1=-4
80 PRINT"QQQQQQQQQQQQQQQQQQQQQQ◤ ▼███◤  ▼█████◤▼███◤ ▼█████◤▼████◤         ▼";
85 PRINT"▒      ▼███◤  ▼█◤  ▼███◤  ▼██◤                  ";
90 PRINT"▒        ▼      ▼       ▼█◤    ▼◤                 ";
95 PRINT"▒                        ▼                      ▒":TA=INT(RND(1)*10)
100 FORI=56256TO56295:POKEI-C0,160:POKEI,9:NEXT
105 FORI=56256+TATO55837+TASTEP-39:POKEI-CO,233:POKEI,5:NEXT
110 FORI=56278+TATO55837+TASTEP-41:POKEI-CO,223:POKEI,5:NEXT
115 FORI=0TO10:FORJ=56257-I*40+I+TATO56277-I*40-I+TA:POKEJ-CO,160:POKEJ,5:NEXT
120 NEXT
125 IFRND(1)<.05ORPEEK(V+30)=2THEND1=-D1:POKE54273,10
130 X1=X1+D1:IFRND(1)<.1THENY1=Y1+((Y1>1)-(Y1<Y))*4:GOTO140
135 Y1=Y1+(INT(RND(1)*3)-1)*4
140 IFX1<10ANDD1=-4THENX1=320
145 IFX1>320ANDD1=4THENX1=0
150 IFY1<60THENY1=60
155 IFY1>200THENY1=200
160 IFRND(1)>.2ORY1<Y-10ANDY1>Y+10ANDRND(1)<.30ORS2<>0ORX1>280THEN175
165 S2=X1+(D1*5):D2=D1*3:IFD1=4THENS2=S2+30:POKEV+7,Y1
170 FORI=50TO0STEP-2:POKE54273,I:NEXT:POKE54273,0
175 IFD1=-4THENPOKE2041,193
180 IFD1=4THENPOKE2041,192
185 POKEV+3,Y1:IFX1>255THENPOKEV+2,X1-256:POKEV+16,PEEK(V+16)OR2:GOTO195
```

PROGRAMA commodore Magazine COMPROBADO

Viene de la página anterior

```
190 POKEV+2,X1:POKEV+16,PEEK(V+16)AND253
195 IFPEEK(V+30)=3THEN340
200 PE=PEEK(197):IFPEEK(V+30)=6THEN360
205 IFPE=9ANDY>60THENY=Y-4:POKE54273,200
210 IFPE=23THENY=Y+4:POKE54273,200
215 IFPE=10THEND=-4:POKE54273,10
220 IFPE=18THEND=4:POKE54273,10
225 IFPE<>13ORSX<>0ORX>200THEN240
230 SX=X+(D*5):POKEV+5,Y:SD=D*3:IFD=4THENSX=SX+30
235 FORI=50TO0STEP-2:POKE54273,I:NEXT:POKE54273,0
240 X=X+D:IFD=4THENPOKE2040,192
245 IFD=-4THENPOKE2040,193
250 IFX<10ANDD=-4THENX=320
255 IFX>320ANDD=4THENX=0
260 POKE54273,0:POKEV+1,Y:IFX>255THENPOKEV,X-256:POKEV+16,PEEK(V+16)OR1:GOTO270
265 POKEV,X:POKEV+16,PEEK(V+16)AND254
270 IFPEEK(V+300)=3THEN340
275 IFPEEK(V+30)=9THEN340
280 IFPEEK(V+31)=1THEN340
285 IFSX=0THEN310
290 IFSX<10ORSX>310THENSX=0:SD=0:POKEV+5,0:POKEV+4,50
295 SX=SX+SD:IFSX>255THENPOKEV+4,SX-256:POKEV+16,PEEK(V+16)OR4:GOTO305
300 POKEV+4,SX:POKEV+16,PEEK(V+16)AND251
305 IFPEEK(V+30)=6THEN360
310 IFS2=0THEN125
315 IFS2<10ORS2>310THENS2=0:D2=0:POKEV+7,0:POKEV+6,230
320 S2=S2+D2:IFS2>255THENPOKEV+6,S2-256:POKEV+16,PEEK(V+16)OR8:GOTO330
325 POKEV+6,S2:POKEV+16,PEEK(V+16)AND247
330 IFPEEK(V+30)=9THEN340
335 GOTO125
340 POKEV+6,0:POKEV+7,0:FORI=YTO200:POKEV+1,I:POKE54273,I
345 FORJ=1TO5:NEXT:NEXT:POKE54273,0
350 FORI=1TO7:POKEV+I,0:NEXT
355 GOTO480
360 POKEV+4,0:POKEV+5,0:FORI=Y1TO200:POKEV+3,1:POKE54273,I
365 FORJ=1TO5:NEXT:NEXT:POKE54273,0
370 FORI=1TO7:POKEV+I,0:NEXT
375 GOTO530
380 PRINT"■■|||||||||■DUELO EN LAS ESTRELLAS■|||||||";
385 FORI=1TO11
390 PRINT"||||||||||||||||||||||||||||||||||||||||";
395 NEXT
400 POKEV+8,165:POKEV+9,100
405 PRINT"|■ SOY RAM DEL PLANETA BYTE. TE HE ELEGIDO A TI PARA QUE"
410 PRINT"                    ■DEFIENDAS TU PLANETA."
415 PRINT"■YO TE DESAFIO A LUCHAR!"
420 PRINT"ACEPTAS? S/N"
425 GETA$:IFA$="N"THEN PRINT "■COBARDE":END
430 IFA$<>"S"THEN425
435 PRINT"  EL ENFRENTAMIENTO TENDRA LUGAR AL"
440 PRINT"NORTE DEL PLANETA MARTE"
445 PRINT"    ■DEBO ADVERTIRTE QUE SOLO UNO DE TUS     DISPAROS PUEDE DESTUIRME."
450 PRINT"■  TU PUEDES ESTRELLARTE!"
```

Programas

```
455 PRINT"▤Q PULSA ´ESPACIO´ CUANDO ESTES PREPARADO."
460 GETA$: IFA$<>" "THEN460
465 PRINT"▯":POKEV+9,0
470 POKE53281,0:POKE53280,0
475 RETURN
480 POKE53280,4:POKE53281,4:PRINT"▯":POKE54273,0
485 PRINT"▤";:FORI=1TO12
490 PRINT"||||||||||||||||||||||||||||||||||||||||";
495 NEXT (FBLD)
500 PRINT"■ADIOS MORTAL.LUCHASTE BIEN."
505 PRINT"▤QQQ    ALGUN OTRO HUMANO SE ATREVE A          RETARME? S/N
510 GETA$:IFA$="S"THENGOSUB465:GOTO75
515 IFA$<>"N"THEN510
520 PRINT"■MUY BIEN, LA TIERRA PRONTO SERA MIA!"
525 END
530 POKEV+23,16:POKE53280,4:POKE53281,4:POKEV+8,150:POKEV+9,100:POKE54273,0
535 POKE54296,0:POKE54296,15:POKE54276,0:POKE54276,17
540 PRINT"▤QQQQQQQQQQQQQQQESTOY HERIDO TERRICOLA!"
545 FORI=1TO200:POKE54273,I:NEXT:POKEV+9,0:FORI=100TO1STEP-.5
550 POKE54273,I:NEXT:FORI=10TO50STEP.5
555 POKE54273,I:NEXT:POKEV+23,0:POKEV+29,0:POKEV+8,150:POKEV+9,100
560 FORI=1TO50STEP.5:POKE54273,I:NEXT:POKEV+8,0:POKEV+9,0:FORI=20TO1STEP-.2
565 POKE54273,I:NEXT:POKE54273,0
570 PRINT"■▤ADIOS!!!!"
575 GETA$:FORI=1TO5000:NEXT:PRINT"▯"
580 GOTO70
585 END
590 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,3,0,0,3,192,0,3,192,0,19,240,0
595 DATA7,176,0,67,247,128,7,183,32,83,247,245,2,166,245,2,240,0
600 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,99
605 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,192,0,0,192,0,3,192,0,3,192,0,15,196
610 DATA0,14,208,2,223,193,8,222,208,95,223,197,95,218,128,0,15,128,0,0,0,0,0
615 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,99
620 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,4,0
625 DATA0,21,0,0,127,64,0,21,0,0,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,99
630 DATA60,0,60,12,170,48,15,170,240,14,170,176,2,130,128,2,0,128,2,60,128
635 DATA10,60,160,10,0,160,10,0,160,8,0,32,8,170,32,8,150,32,8,150,32,88,170
640 DATA37,88,0,37,90,0,165,21,170,84,21,130,84,5,170,80,5,170,80
645 PE=PEEK(197):IFPE<>1THENRETURN
650 GOTO650
```

PROGRAMA COMPROBADO commodore Magazine

¡Sorteamos 500 cassettes de juegos para VIC-20 y C-64!

Como podréis suponer, nuestra máxima ambición sería conoceros a todos personalmente, saber vuestro interés y ambiciones informáticos. Pero desgraciadamente eso no es posible más que en casos contados.

De momento sabemos que la revista en grandes líneas os gusta. Eso es lo que demostráis mes a mes en los kioskos y con las tarjetas de suscripción.

No obstante es factible tener un conocimiento amplio de vuestras demandas, por un medio frío pero práctico: la antipática encuesta. Queremos conocer vuestras opiniones, qué ordenador tenéis, etc. Con ese mayor conocimiento confiamos en ir mejorando cada día los contenidos de **Commodore Magazine**. Os pedimos que seáis tan amables de tomaros esta pequeña molestia, estamos seguros que su fruto lo notaréis en breve. Somos conscientes de que todo esfuerzo, por pequeño que sea, debe llevar emparejada una compensación. Quinientas cintas de cassette sorteadas entre todos los que respondan nos parece un estímulo interesante. Así que ¡ánimo!

No olvidéis nuestra dirección:

commodore Magazine

C/ Bravo Murillo, 377, 5.° A.
28020.-Madrid

ENCUESTA

1) ¿Tiene ordenador?
a) VIC-20 .... □
b) C-64 .... □
c) Otro .... □

Díganos su marca ....

2) ¿Dónde adquirió su ordenador?
a) Grandes almacenes .... □
b) Tienda especializada .... □
c) Regalo familiar .... □
d) Otros .... □

3) ¿Para qué utiliza su ordenador?
a) Juegos .... □
b) Educación .... □
c) Aplicación .... □

4) ¿Desde cuándo posee su ordenador?
a) Más de 1 año .... □
b) Más de seis meses .... □
c) Más de 1 mes .... □

5) ¿Cuándo lo utiliza?
a) Todos los días .... □
b) Fin de semana .... □
c) Ocasionalmente .... □

6) ¿Con qué frecuencia lee Commodore Magazine?
a) Todos los números .... □
b) Ocasionalmente .... □
c) Soy suscriptor .... □

7) Indique un orden de preferencia entre los temas de la revista:
a) Programas .... □
b) Artículos de divulgación .... □
c) Temas técnicos .... □
d) Cartas al director .... □
d) Otros ....
....
....
....

8) ¿Qué le gustaría encontrar en Commodore Magazine?
a) ....
b) ....
c) ....

9) ¿En qué estrato de edad se encuentra?
a) Menor de 13 .... □
b) Entre 13 y 18 .... □
c) Entre 18 y 25 .... □
d) Entre 25 y 50 .... □
e) Más de 50 .... □

10) ¿Cuál es su ocupación habitual?
....
....

11) Añada aquí sus comentarios
....
....
....
....

CUANDO SE TIENE UN COMMODORE 64 ES MUY DIFICIL SER MODESTO

Cuando se tiene un ordenador personal con 64K de memoria, una magnífica resolución, 16 colores, efectos tridimensionales con "sprites", un sonido equivalente al de un sintetizador, un teclado profesional con 62 caracteres gráficos, toda una amplia gama de periféricos profesionales, la más completa serie de programas educativos, profesionales y de video-juegos...; en resumen, cuando se tiene un ordenador personal como no existe ningún otro en el mercado y el más vendido mundialmente, cuando se tiene el Commodore 64, es muy difícil mostrarlo sin que el orgullo se te note.



EL ORDENADOR PERSONAL DE LA FAMILIA MAS POTENTE

– Sistemas de gestión profesionales series 8000 Y 700. – Ordenador portátil SX 64.
– Ordenador personal COMMODORE 64. – Ordenador familiar VIC 20.



SLOGAN

MICROELECTRONICA Y CONTROL, S.A.
c/ Taquígrafo Serra, 7, 5.° BARCELONA-29 c/ Princesa, 47, 3.° G MADRID-8